

QR ATTENDANCE SYSTEM

**Report II, literature review, system analysis and design, and
prototype Implementation**

**A report submitted to the
Faculty of Engineering**

**In partial fulfillment of the requirements for the degree of
Bachelor of Science
In Computer Engineering**

Prepared by:

Student Name	ID
Ahmed El Sayed	51610435
Andrew Atef	51610451
Mahfouz Assem	51610491
Mahmoud El Telbany	51610295
Mohamed Mostafa	51610240
Mostafa Mahdy	51610455
Yara Hossam	51610330
Youssef Ahmed	51610425

Supervised by

Dr. Mohamed ElAskely & Dr. Soha Mohamed

Faculty of Engineering, Ahram Canadian University

2020/2021

TABLE OF CONTENTS:

TABLE OF CONTENTS.....	1
CHAPTER 1.....	7
INTRODUCTION	7
1.1 HISTORY OF QR CODES	7
1.2 THE DIFFERENCE BETWEEN QR CODE AND BARCODE	8
1.3 QR CODE APPLICATION.....	9
1.4 AUTOMATIC ATTENDANCE SYSTEMS USING QR	11
1.4.1 ATTENDANCE MONITORING AND MANAGEMENT USING QR CODE BASED SENSING WITH CLOUD BASED PROCESSING [SINGH, 2019]	11
1.4.2 SMART MOBILE ATTENDANCE SYSTEM FOR EMPLOYEES USING QR SCANNER [KUMAR, 2017]	13
1.4.3 Attendance checking System Using Quick Response Code for students at university of Sulaimaniyah [Babani, 2014].....	13
1.4.4 A student attendance system using QR code [Almasalha, 2014]	14
1.4.5 Location-aware Event Attendance System using QR Code and GPS Technology [Ayoub, 2018].....	16
1.4.6 QR code based smart attendance system [Wei, 2017].....	17
CHAPTER 2.....	21
Attendance system methodologies and the employed programming platforms.....	21
2.1 INTRODUCTION.....	21
2.2 METHODOLOGY	21
2.2.1 Incremental Model.....	22
2.2.1.1 Requirement Analysis.....	23
2.2.1.2 Context Model.....	24
2.2.1.3 Use Case Model.....	25
2.2.1.4 Sequence Model.....	26
2.2.1.5 Data Flow Model.....	27
2.2.1.6 The Overall System Structure and System Flowcharts.....	28
2.3 SOFTWARE PLATFORMS	32
2.3.1 MOBILE OPERATING SYSTEMS	32
2.3.2 Website Login.....	32
2.3.3 Database.....	32
2.3.3.1 MongoDB.....	32
2.3.3.2 Firebase.....	33
2.3.3.3 PhpMyAdmin.....	33
2.3.3.4 PHP.....	33
2.3.3.5 MySQL.....	34
2.4 DATABASE IN OUR PROJECT	34

2.5 THE WEBSITE IN OUR PROJECT.....	38
2.6 COVID-19 PANDEMIC CHALLENGING AND MOTIVATION	43
2.7 Future Work.....	44
CHAPTER 3.....	45
PRACTICAL WORK AND USER INSTRUCTIONS	45
3.1 WEBSITE PART	45
3.1.1 QR-Code Generation	45
3.1.2 Student Submission	51
3.2 Mobile Application Part.....	61
3.2.1 Authentication In Mobile Application.....	61
3.2.2 Student History In Mobile Application.....	64
3.2.3 Courses History In Mobile Application.....	65
3.2.4 Controlling Pages In Mobile Application.....	67
PROJECT PLANNING REPORT	74
REFERENCES:.....	78

LIST OF FIGURES:

Fig. 1.1: QR code vs barcode images.....	9
Fig. 1.2: System flow Diagram.....	12
Fig. 1.3. System architecture.....	13
Fig. 1.4. Excel sheet for checking class attendance.....	14
Fig. 1.5. The proposed system infrastructure.....	15
Fig. 1.6. Mobile Module Flow Chart.....	15
Fig. 1.8. User View Flowchart.....	17
Fig 1.9. Admin View Flowchart.....	18
Fig. 1.10 Use case diagram.....	19
Fig.1.11 Data Flow Diagram.....	20
Fig 1.12. Flowchart of the application system.....	20
Fig. 2.1: Incremental Model.....	23
Fig. 2.2: Attendance system context model.....	26
Fig. 2.3: Attendance system use case diagram.....	26
Fig. 2.4: Attendance system sequence diagram.....	27
Fig. 2.5: Attendance system data flow diagram.....	28
Fig. 2.6: The overall structure of attendance system.....	29
Fig. 2.7: Instructor instructions flowchart.....	30
Fig. 2.8: Student instructions flowchart.....	31
Fig. 2.9: Proposed Solution Architecture Design.....	33
Fig. 2.10: Creating database using XAMPP.....	35

Fig. 2.11: Creating a table in the database.....	35
Fig. 2.12: Assigning the characteristics of the table.....	36
Fig. 2.13: Table shape in database.....	36
Fig. 2.14: Uploading excel sheet to database.....	37
Fig. 2.15: Uploading excel sheet to database.....	37
Fig. 2.16: How the data in the database looks like.....	38
Fig. 2.17: HTML code example.....	39
Fig. 2.18: CSS code example.....	39
Fig. 2.19: JavaScript QR code example.....	40
Fig. 2.20: JavaScript location code example.....	41
Fig. 2.21: JavaScript code example.....	42
Fig. 2.22: Fronted QR code generator.....	42
Fig. 2.23: Data represented from the QR code.....	43
Fig. 3.1: Instructor login home page.....	46
Fig. 3.2: Instructor PHP code for login function.....	46
Fig. 3.3: Current lecture attendance creation page.....	47
Fig. 3.4: Forgetting to delete last week attendance.....	48
Fig. 3.5: Table creation PHP code.....	48
Fig. 3.6: current lecture attendance table and overall semester attendance tables in database.....	49
Fig. 3.7: QR-code generating page in addition to different attendance reports.....	50
Fig. 3.8: QR-code generating page in case of QR-code generating.....	50

Fig. 3.9: Student submitting page.....	51
Fig. 3.10: Student submitting page creation PHP code.....	52
Fig. 3.11: QR-code.js page.....	53
Fig. 3.12: JavaScript functions.....	53
Fig. 3.13: Attendance denying message for twice registry from the same mobile.....	54
Fig. 3.14: Preview of the lecture attendance table in the database.....	54
Fig. 3.15: Students' attendance reports.....	55
Fig. 3.16: Semester total attendance table.....	56
Fig. 3.17: Semester view table attendance counter.....	56
Fig. 3.18: Attendance list HTML/PHP creation code.....	57
Fig. 3.19: Downloading different attendance reports.....	58
Fig. 3.20: Downloading the excel sheet.....	58
Fig. 3.21: Excel sheet contains the count of total attendance for each student.....	58
Fig. 3.22: PHP code for today's Attendance table download button.....	59
Fig. 3.23: PHP code for semester Attendance export button.....	60
Fig. 3.24: Continue of PHP code for semester Attendance export button.....	61
Fig. 3.25: Authentication database in Firebase.....	62
Fig. 3.26: Cloud Firestore database structure.....	64
Fig. 3.27: Document for every student ID.....	65
Fig. 3.28: Show location.....	65

Fig. 3.29: Student QR-code scanning history.....	66
Fig. 3.30: Courses document.....	66
Fig. 3.31: History of courses.....	67
Fig. 3.32: Mobile application login page.....	68
Fig. 3.33: Main page of mobile application after login.....	68
Fig. 3.34: Mobile application user profile page.....	69
Fig. 3.35: User history page one.....	69
Fig. 3.36: User history page two.....	70
Fig. 3.37: Admin dashboard page.....	70
Fig. 3.38: Access of student data by admin login.....	71
Fig. 3.39: Courses information page.....	71
Fig. 3.40: Course history contains students scanned QR code.....	72
Fig. 3.41: The history of student's information scanned QR code.....	72
Fig. 3.42: The history of users scanned QR code info.....	73

Chapter 1

INTRODUCTION

In most of academic facilities, a faculty member prepare students attendance file that contains all student list for both theory and lab before starting classes. For taking attendance, the instructor call students name one by one and mark their attendance in a paper. All these steps are done manually which wastes both students and instructors valuable time. Moreover, the COVID 19 pandemic raised the need for a contactless life style. All of this has led to the creation of the automated attendance systems.

Automated attendance system is a powerful solution that ensures maintaining attendance with reducing paperwork and saving time and money. The system improves visibility to track and manage student attendance & absenteeism with automatic calculation of attendance days. Automation facilitates generation of various types of class reports. Moreover, parents might be informed about the student's performance via Email & SMS alerts.

Different technologies have been adopted to automate the process of taking attendance. Al Hajri, et.al. depended on the infrared technology in their approach [Al Hajry, 2019]. Gupta and his colleagues proposed an idea which is completely based on face detection technique where the system generates the spreadsheet which provides the no. of students present in classroom with an image or video capturing live [Gupta, 2020]. Other approaches consider QR codes technology and these will be discussed thoroughly later.

1.1. History of QR codes

QR codes owe their existence to the development and success of barcodes. Since their creation, barcodes became very popular due to the speed at which they could be scanned, the accuracy they provided, and their multiple functionalities. With increased popularity, there was growing demands for barcodes that could store more information, had more variation, and would take up a smaller printing area.

Many efforts were made to improve the barcode technology and how much information they could store. These efforts included increasing the number of digits in the codes, modifying the

layout to include multiple codes and more. However, these modifications came with drawbacks such as increasing the size of the barcode and printing space it needed. Moreover, some development complications occurred while attempting to read/scan the new codes. Therefore, the 2D (two-dimensional) QR codes were the response to provide more storage capacity, functionality, accuracy, etc. As with the development of barcodes, since their creation QR codes have seen increasing population, developments in the technology to improve storage capacity and functionality.

(Quick Response) QR codes were first created back in 1994 [Chang, 2014]. A Toyota subsidiary named Denso Wave developed the code in order to help in the manufacturing process where they aided in tracking vehicles and parts. It was designed to allow for fast decoding speeds, hence the name Quick Response code.

1.2. The difference between QR code and Barcode

The main difference between QR code and barcode is represented in the capacity of information that each one can hold. While a barcode only holds information in the horizontal direction (one dimensional code), a QR code does hold information in both horizontal and vertical directions. Due to this, a QR code holds hundreds of times more information than a barcode. Fig. 1 shows an example of the shape for both QR code and barcode, respectively.

A barcode is a visual representation of machine-readable information about the product to which it's attached. It is that small box printed on the packaging of a product that has a small collection of black parallel lines of varying widths. It contains important information like the manufacturer's name, type of item, price and so on, which can only be read by dedicated barcode reader machines. Since it only consists of data stored in horizontal direction, it is also dubbed as linear or 1D (1-dimensional).

On the other hand, QR code is much similar to a barcode. It also contains machine-readable information concerning the item to which it is attached. But, unlike a regular barcode, a QR code is two-dimensional, i.e., it contains information both in vertical and horizontal directions. Hence, QR codes contain a lot of information within them. From giant corporate organizations to next-door grocery store, everyone can create their own QR code and attach them to their products. QR codes have gained popularity during the last decade due to the rise of

smartphones whereas certain QR code scanner apps can be downloaded for free on smartphones. So, the moment you see a QR code, just scan it using the app on your phone (the app generally uses the primary camera of the phone to scan the code) and the pre-determined action (like opening a website, sending a tweet, download an app etc.) attached to the code will be triggered.



Fig. 1.1: QR code vs barcode images

QR Code applications are now on the rise more than ever as it provides a contactless solution for most operations such as contactless payment, contactless delivery, and even contactless retail. Smartphones today have the capability to scan even damaged QR Codes, but the main problem of QR code arises in the possibilities that it might not be scanned. There are mainly 6 reasons why a QR Code does not work:

- Bad quality.
- Inverted colors.
- Small size.
- Poor contrast ratio.
- Wrong Placement.
- Overcrowded QR Code.

1.3. QR code applications

Although initially used to track parts in vehicle manufacturing, QR codes are used over a much wider range of applications. These include commercial tracking, entertainment and transport ticketing, product and loyalty marketing and in-store product labeling. For example, in marketing applications, a company's discounted and percent discount can be captured using a QR code decoder or storing a company's information such as address and related information alongside its alpha-numeric text data as can be seen in Yellow Pages directory.

Moreover, QR codes have become common in consumer advertising. Typically, smartphones are used as a QR code scanner, displaying the code and converting it to some useful form (such as a standard URL for a website, thereby obviating the need for a user to type it into a web browser). QR code has become a focus of advertising strategy, as it provides a way to access a brand's website more quickly than by manually entering a URL. Beyond mere convenience to the consumer, the importance of this capability is that it increases the conversion rate: the chance that contact with the advertisement will convert to a sale. It coaxes interested prospects further down the conversion funnel with little delay or effort, bringing the viewer to the advertiser's website immediately, whereas a longer and more targeted sales pitch may lose the viewer's interest.

They can also be used in storing personal information for use by organizations. Many of these applications target mobile-phone users (via mobile tagging). Users may receive text, add a vCard contact to their device, open a URL, or compose an e-mail or text message after scanning QR codes. They can generate and print their own QR codes for others to scan and use. As an example of this application, the Philippines National Bureau of Investigation (NBI) issues NBI clearances with a QR code.

QR codes storing addresses and URLs may appear in magazines, on signs, on buses, on business cards, or on almost any object about which users might want information. Users with a camera phone equipped with the correct reader application can scan the image of the QR code to display text, contact information, connect to a wireless network, or open a web page in the telephone's browser. This act of linking from physical world objects is termed hard linking or object hyperlinking. QR codes also may be linked to a location to track where a code has been scanned. Either the application that scans the QR code retrieves the geo information by using GPS and cell tower triangulation (AGPS) or the URL encoded in the QR code itself is associated with a location. In 2008, a Japanese stonemason announced plans to engrave QR codes on gravestones, allowing visitors to view information about the deceased, and family members to keep track of visits. Psychologist Richard Wiseman was one of the first authors to include QR codes in a book, in *Paranormality: Why We See What Isn't There* (2011).

QR codes have been incorporated into currency. In June 2011 The Royal Dutch Mint (Koninklijke Nederlandse Munt) issued the world's first official coin with a QR code to celebrate

the centenary of its current building and premises. The coin can be scanned by a smartphone and originally linked to a special website with contents about the historical event and design of the coin. In 2014 the Central Bank of Nigeria issued a 100-naira banknote to commemorate its centennial, the first banknote to incorporate a QR code in its design. When scanned with an internet-enabled mobile device, the code goes to a website which tells the centenary story of Nigeria. In 2015, the Central Bank of the Russian Federation issued a 100-rubles note to commemorate the annexation of Crimea by the Russian Federation. It contains a QR code into its design, and when scanned with an internet-enabled mobile device, the code goes to a website that details the historical and technical background of the commemorative note. In 2017, the Bank of Ghana issued a 5-cedis banknote to commemorate 60 years of Central Banking in Ghana, and contains a QR code in its design, which when scanned with an internet-enabled mobile device, that code goes to the official Bank of Ghana website.

Credit card functionality is under development. In September 2016, the Reserve Bank of India (RBI) launched the eponymously named Bharat QR, a common QR code jointly developed by all the four major card payment companies - National Payments Corporation of India that runs RuPay cards along with MasterCard, Visa and American Express. It will also have the capability of accepting payments on the unified payments interface (UPI) platform.

1.4. Automatic attendance systems using QR

As afore mentioned, taking attendance using the traditional way is a very slow process that usually takes a lot of lecture's time. Different approaches were proposed to automate this process. This section surveys different proposed approaches to automate the process of taking attendance by using QR codes.

1.4.1 Attendance monitoring and management using QR code based sensing with cloud based processing [Singh, 2019]

The idea of this approach relies on the fact that each student has a smartphone with an android application. This application enables the students to register their attendance and know how many attendance days in a month they have registered. In this system, the instructor can log in to calculate the number of attendance day for each student and sent a report of their attendance and absence days to their parents by mail if the attendance was less than 50%.

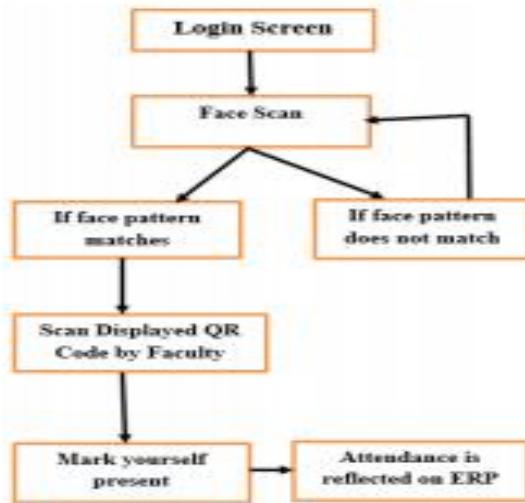


Fig. 1.2: System flow Diagram

The proposed system consists mainly of two modules; (i) the instructor module and (ii) the student module. The instructor module is a web based application that displays the QR code, edit the students' attendance, and calculate and send attenuation to students. On the other hand, the student module is an android application that scans the QR code to mark the attendance in lectures. As shown in Fig 1.2, the system registers attendance for students based on two factors: (i) The face recognition, as the face recognition is used to log in to the application only by matching the pattern of the captured face with the previously stored and (ii) the QR code, which plays the most important role in this system. The QR code is created by the faculty and will be available for a specific time during which the student can open the application and scan the QR code to place a mark on his name.

1.4.2 Smart mobile attendance system for employees using QR scanner [Kumar, 2017]

Similar to the previous approach, Kumar, at.al., combine the QR code with biometrics technologies. The biometrics includes different techniques to check the identity of the person like finger prints, face ID, iris, and retinal patterns. Human face recognition (HFR) is one of the known techniques which can be used for user authentication. Using HFR is usually adopted to eliminate the need to remember a password, as in traditional methods. This approach starts by capturing a sample of a biological feature that acts as an effective way to upload students' attendance themselves by accessing the login to the attendance portal and by scanning relevant QR codes for various subjects.

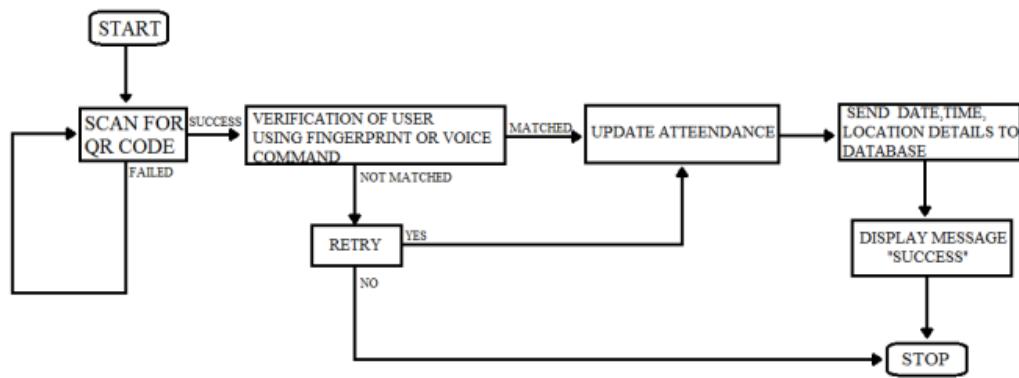


Fig. 1.3. System architecture

Fig. 3 shows the system architecture that depends on installing the attendance application on a smartphone. After logging in to the application, the user scans the QR code. After the scanning process is completed successfully, he/she is connected to the database, otherwise, he must scan it again. The user is required to verify his/her identity by using biometric features such as fingerprint or voice command. Then, if the verification is successful, attendance is confirmed, and if it fails, he is given another opportunity and then logged out. This is done after verifying the date, time, and geographical location.

1.4.3. Attendance checking system using Quick Response code for students at the university of Sulaimaniyah [Baban, 2014]

Baban proposed an incomplete system that needs a lot of manual work. This system displays the average weekly absence of students. This work utilized the traditional of taking

attendance where each teacher will receive an Excel file and every time he/she takes the absence verbally or in writing, he/she will determine the names of the absent students. Then, the file is sent to the administrator to be entered into the database. In this system, the QR code is generated by the teacher that leads to an URL link to enable students to check the percentage of their absence.

Fig. 1.4. Excel sheet for checking class attendance

1.4.4. A student attendance system using QR code [Almasalha, 2014]

In their work, Almasalha and Hiezallah divided the system into two modules. The first module, the server, creates the QR code and performs both the identity and the location checks, as shown in Fig. 1.5. The server sends the student's ID, the date and time of the lecture, the attendance status, and a small-sized image of the student's face to the educational platform where the server acts as an intermediary to record attendance.

The second module is a mobile module that the students usually install on their mobile phones. As depicted in Fig. 1.6, once the student sees the QR code on the screen, he/she opens the Mobile application. If it is the first time after restarting the Smartphone, the system requests the student to enter a username and password. Once logged in, the system prompts the student to click on the start button. The system will then capture the face of the student. The facial image will be checked against standard facial conditions, such as locating the eyes. Once the image is accepted, the system requests the user to scan a QR code within a very short time. Once the code is scanned, the system sends the information to the server and resumes working in the background. With that, the process is considered completed. The server in turn will send back an acknowledgment that the process is complete.

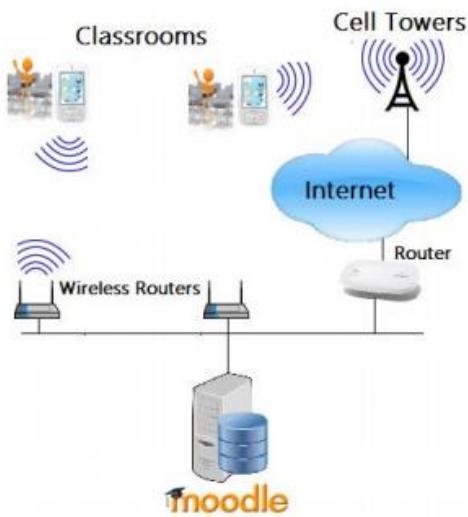


Fig. 1.5. The proposed system infrastructure

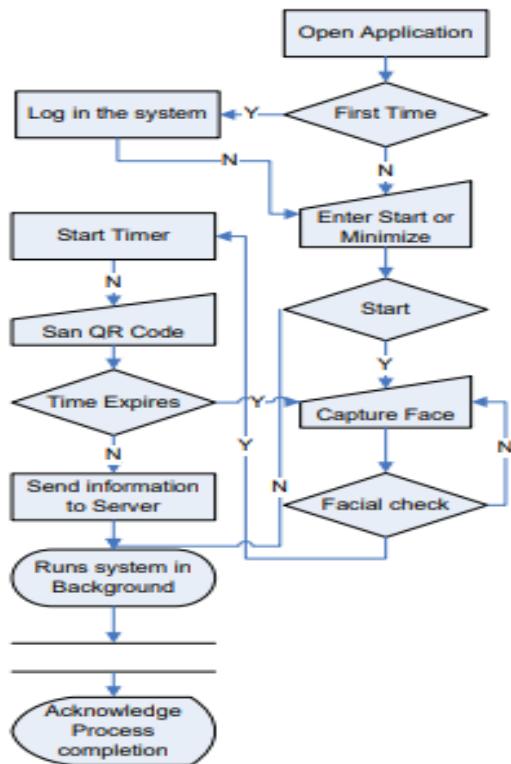


Fig. 1.6. Mobile Module Flow Chart

1.4.5. Location-aware Event Attendance System using QR Code and GPS Technology [Ayoub, 2018]

The system traces the attendance of students using the GPS location combined with the QR code. The results indicate that full attendance increases as the system validates attendance during user login and logout by identification of users, location and timestamp. The system development process is based on two views; (i) the user view, which is the students' smartphone application, and (ii) the admin view, which is the event organizer's web administration system. Both views are shown in Fig. 1.8 and Fig 1.9, respectively

A. User view

The user view flow chart for the Event Attendance System is shown in Fig. 1.7. A student can login to the system with the android application. The student will select the "SCAN THE QR CODE" button to scan the QR code created by the event organizer of the university after logging into the system successfully. The data about the event that is included in the QR code, the location of the student, and student identification will be submitted to the database server after the scanning process. This is to ensure that when he/she reports their presence, the student is inside the event hall / location. To have the attendance saved in the database, the student has to scan the QR code 15 minutes before the event starts, and has to log out within 15 minutes after the event finishes. The user view mobile application includes four modules: (i) Login, (ii) QR code Scanner, (iii) GPS and (iv) attendance modules. The student information such as username, name, faculty, course and login time stored in the database will be displayed if they are successfully logged into the system. The student needs to press the "SCAN THE QR CODE" if he/she is in the correct location to attend.

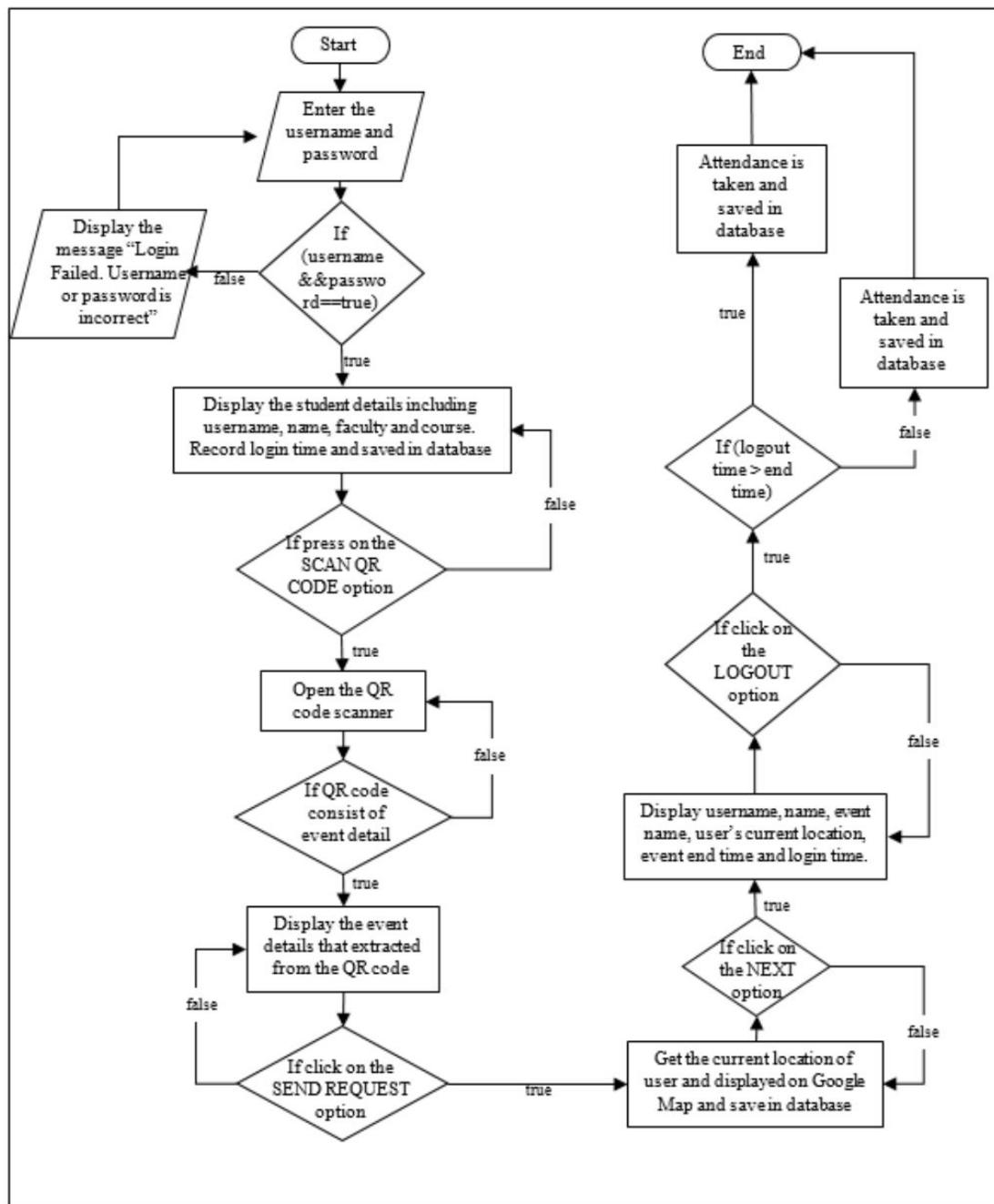


Fig. 1.8. User View Flowchart

B. Admin view

Fig.1.9 shows the flow chart of the admin view. The system administrator is capable to choose different options in the main menu page. If administrator selects the Student Attendance option, he/she can search an event name for viewing all students' attendance who participated in the

event. In addition, administrator is allowed to view the particular event details if he/she indicates the Event Details selection. Furthermore, administrator is capable to add a new event details which will be saved in database and generate a QR code that consists of the event data in Add Event and QR code generator respectively. Finally, the administrator will access the student information on the Student Details page according to the matric number searched.

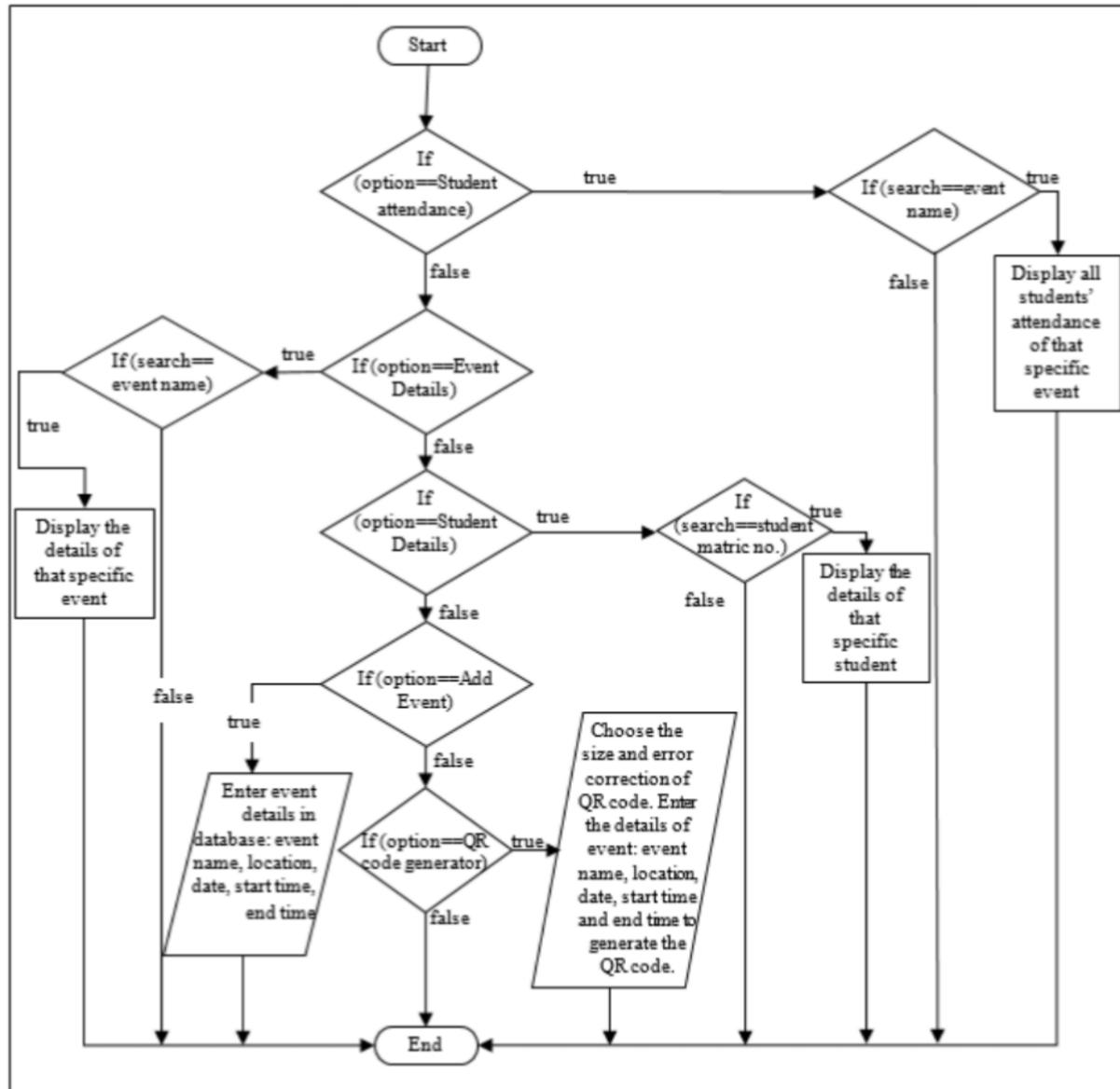


Fig 1.9. Admin View Flowchart

1.4.6. QR code based smart attendance system [Wei, 2017]

As most of attendance systems surveyed, the proposed system consists of two modules, one to generate the QR Code by entering the information of the student and the second to take the attendance and generate the attendance in the form of CSV or XLS. To validate their presence, the instructor would need to check the QR code of the individual student. To remove false registrations, the system should verify student identity. The system deals with the management and evaluation of all students' attendance. The student QR code will be issued to the teacher for their attendance. It is the responsibility of the professor handling the topics to mark the attendance of all students in the group or class. The attendance will be marked as 0 and 1, 0 for absent, and 1 for a present in the database of the particular student row in the table. The student attendance reports will be generated in CSV and XLS sheet for further use. The system is described using its use case, data flow diagram and system flowchart in Fig 1.10, 1.11, and 1.12, respectively.

1.4.6.1. Use Case

The various actors of the system along with their functions are described in Fig. 1.10.

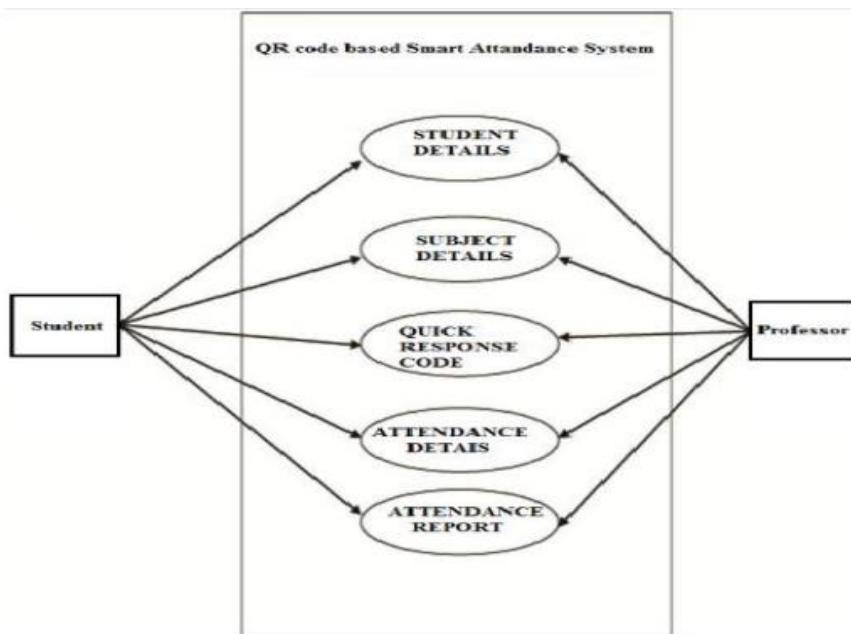


Fig. 1.10 Use case diagram

1.4.6.2. Data Flow Diagram

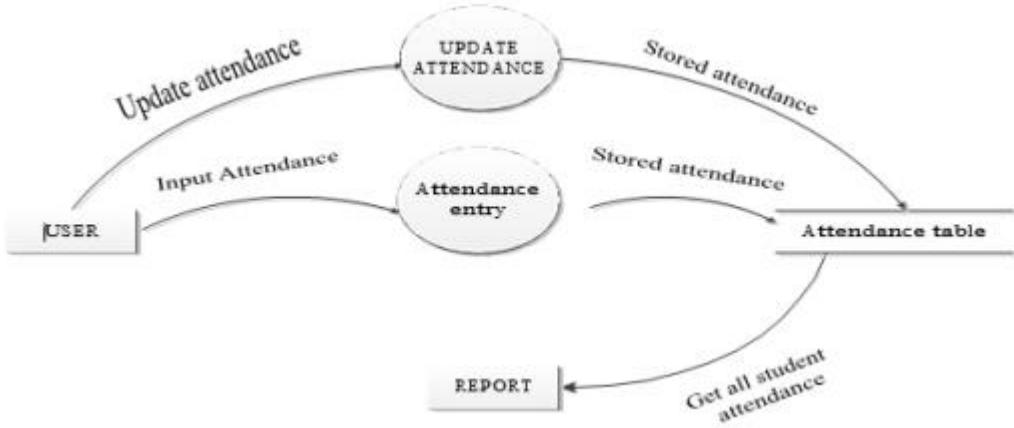


Fig.1.11 Data Flow Diagram

1.4.6.3. System Flowchart

The flow chart of the problem is shown in the figure below.

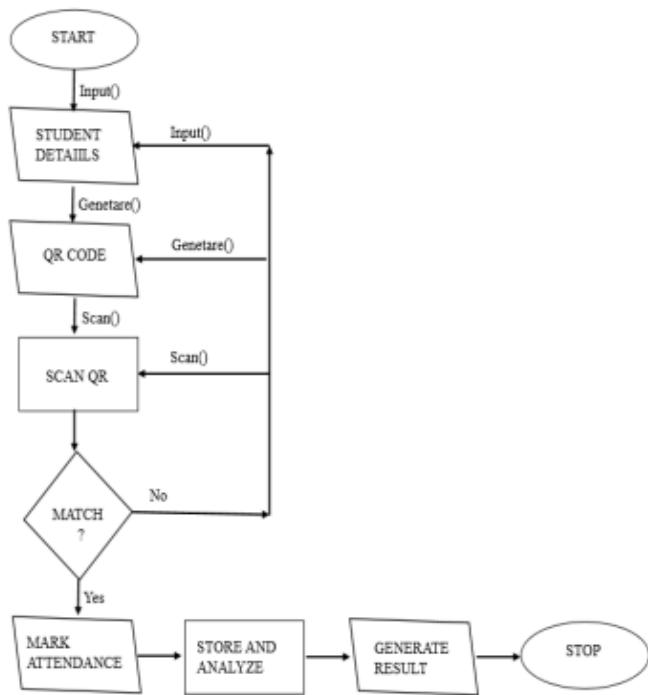


Fig 1.12. Flowchart of the application system

Chapter 2

Attendance system methodologies and the employed programming platforms

2.1 Introduction

In this chapter the proposed attendance system software engineering that granted have been discussed. The goals of that section is showing the rules and constraints that used in the steps of building the system. In addition, these software engineering rules assuring the system readability, extendibility, flexibility, maintainability, dependability, usability, reliability, reusability, and efficient. These rules represent the map of the system production during different steps such as designing, developing, testing, and software maintenance.

In our project, we are dealing with different environments, requirements, and resources. This Varity make us needs to employ different programming platforms to satisfy the system needs with high performance. To reach high performance attendance system there are different programming platforms have been employed. On the other side, this chapter focus on the software platforms that used in the system production. The project is built using Mongo DB, Firebase, PHPmyadmin, PHP, HTML, MySQL, and android. In the next two sections, the chapter will focus on the two main subjects that we are listed.

2.2 Methodology

QR Code Based Attendance Management System: is a combination of two apps:

- 1- Android applications developed for taking and storing the attendance of the students on the daily basis in the college. Here the professor, who is generating the QR-Code, will be responsible to mark the attendance of the students in Database table.
- 2- Web application developed for generating a QR- code from its properties that it is unique and difficult to penetrate its privacy. Each staff will be given a web application that is used for taking attendance and generate the overall attendance status. An accurate report based on student attendance is generated here. Report of the student's attendance on a

weekly and monthly basis is generated as desired. The main objective of the automated attendance system is to computerize the traditional way of recording attendance and provide an efficient and automated method to track attendance in institutions.

Advantages of QR Code Based Smart Attendance System:

- Provide better security.
- Maintenance of the system is easy and cost-effective.
- Generate the result quickly.
- Provide accurate and efficient data.
- User friendly.

To achieve the above-discussed objectives, a step-by-step methodology has been followed. The details of the methodology are given below:

- 1- Develop a QR code generator web app using the details that the teacher who entered.
- 2- Develop an Android app that takes the attendance by scanning QR-Code and generate the student attendance sheet as per attendance details.

2.2.1 Incremental Model

Incremental Model is a process of software development where requirements are broken down into multiple standalone modules of software development cycle. Incremental development is done in steps from analysis design, implementation, testing/verification, maintenance. As shown in figure 2.1.

We use this model because improved and increased user involvement, reduced time and costs and this process can be used with any software developing life cycle model. While this shall be chosen when you are developing a system has user interactions. So, if the system does not have user interactions, such as a system does some calculations shall not have prototypes.

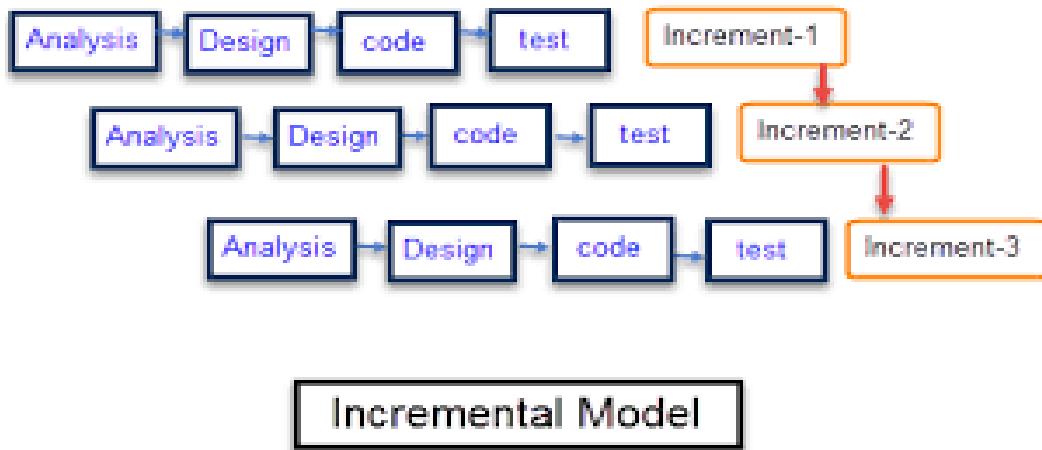


Fig. 2.1: Incremental Model

Incremental Phases	Activities performed in incremental phases
Requirement Analysis	<ul style="list-style-type: none"> Requirement and specification of the software are collected
Design	<ul style="list-style-type: none"> Some high-end function are designed during this stage
Code	<ul style="list-style-type: none"> Coding of software is done during this stage
Test	<ul style="list-style-type: none"> Once the system is deployed, it goes through the testing phase

1.1 WHY WE USED INCREMENTAL MODEL?

- The software will be generated quickly during the software life cycle
- It is flexible and less expensive to change requirements and scope
- Throughout the development stages changes can be done

- This model is less costly compared to others
- A customer can respond to each building
- Errors are easy to be identified

2.2.1.1 Requirement Analysis

1. **User requirements:** user requirements are divided into functional and non-functional requirements:

The user functional requirements for our project is listed as:

- **QR generating:** the lecturer shall be able to generate a QR code
- **QR scanning:** student will be scan QR code
- **QR code generation:** will be done by the lecturer or professor then the student scans it to activate his attendance.
- **Database:** to store the data of each student

The user non-functional requirements for our project is listed as:

Security: is a major issue in our system which we have a lot of things of security such as:

2. **System requirements:** also system requirements are divided into functional and non-functional requirements:
- **Location:** to verify the location of the student to prevent the false attendance
 - **Time:** to determine a certain time for scanning QR code
 - **Smartphone:** to be able to enter to the application of QR code
 - **Application of QR code:** to be able to scan QR code
 - **Characteristics:** Lecturer – student – admin

The system functional requirements for our project is listed as:

- **Professor login:** The user logs in as a teacher with his/her user name and password to be capable to create the event.
- **Student login:** The user logs in as a student with his/her user name and password to be capable to attend themselves.

- **QR scanning for students:** The code scanning application will scan the code and sends information to the database.
- **The system non-functional requirements for our project is listed as:**
- **Usability:** After login, the user can make polling using a mobile phone camera. The attendance can be displayed in list format in DBMS.
- **Reliability:** This system will be dependable to take a short time to confirm the order. The QR code will be randomized every few minutes.
- **Performance:** The software will be based on a local server.
- **Maintainability:** The system is editable; the admin can develop the system at any time to match his requirements.
- **Supportability:** The system supports IOS and android device and any browser with an internet connection.
- **Implementation:**
 1. Website: Xampp, Java, HTML, CSS3, php.
 2. 2-Mobile application: Flutter.
- **Safety and security:** The system data is secured only the admins can access the data and edit them. The system will be secured from students that aren't in the lecture and want to check the attendance while they are absent.
- **Privacy:** Only the admins have access to the software code.

The next section will focus in different software engineering models used in the attendance system production.

2.2.1.2 Context Model

Figure 2.2 shows the different systems that have been established, such as the entry system for students and professors, the security system, the attendance registration system, the database, and how they were interconnected to form an integrated system for us.

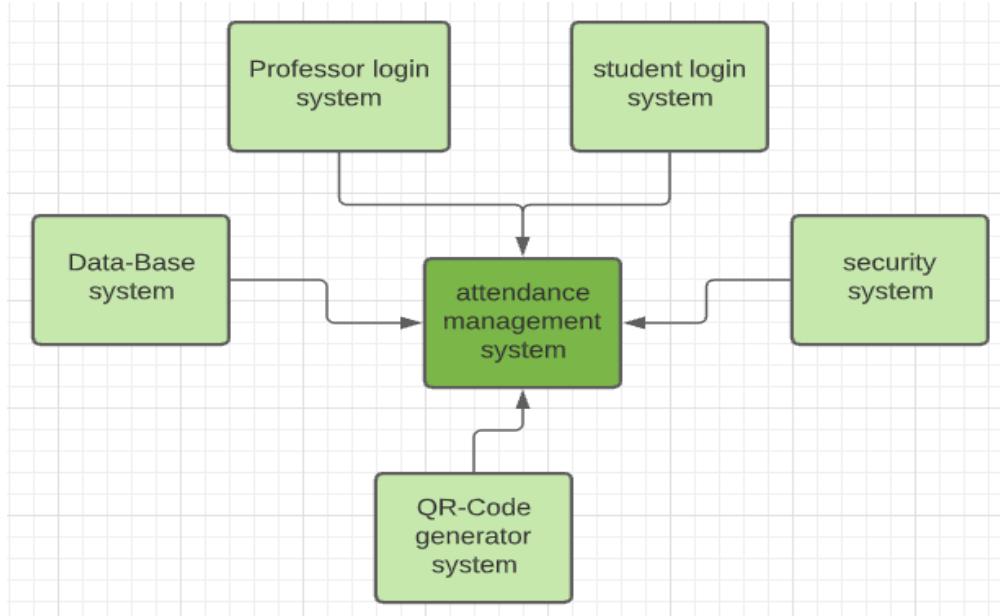


Fig. 2.2: Attendance system context model

2.2.1.3 Use Case Model

The various actors of the system along with their functionality are described in the given figure 2.3, it also explains some of the dealings of each actor in the system and its role.

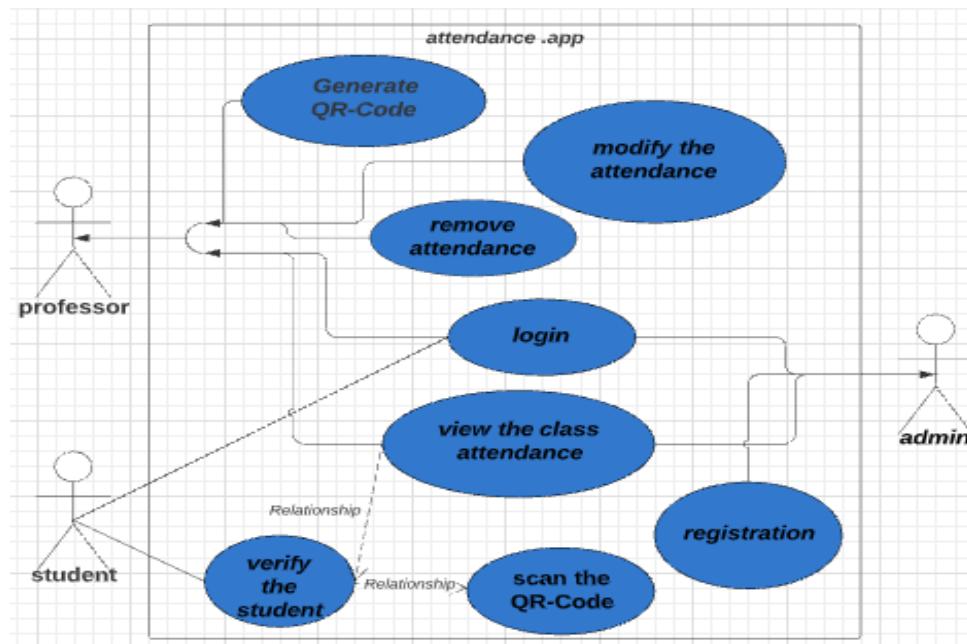


Fig. 2.3: Attendance system use case diagram

2.2.1.4 Sequence Model

The overall sequence diagram of the attendance system is given below. To give a simple summary of figure 2.4, the student enters the application, where the data is entered to enter, and then it is confirmed from the database if it is wrong, then the page is returned to the login again and asked to try again, and if it is correct, the student is allowed to enter and scan the QR-Code, and it is recorded. The student attends, after which the system will return the student to the login page.

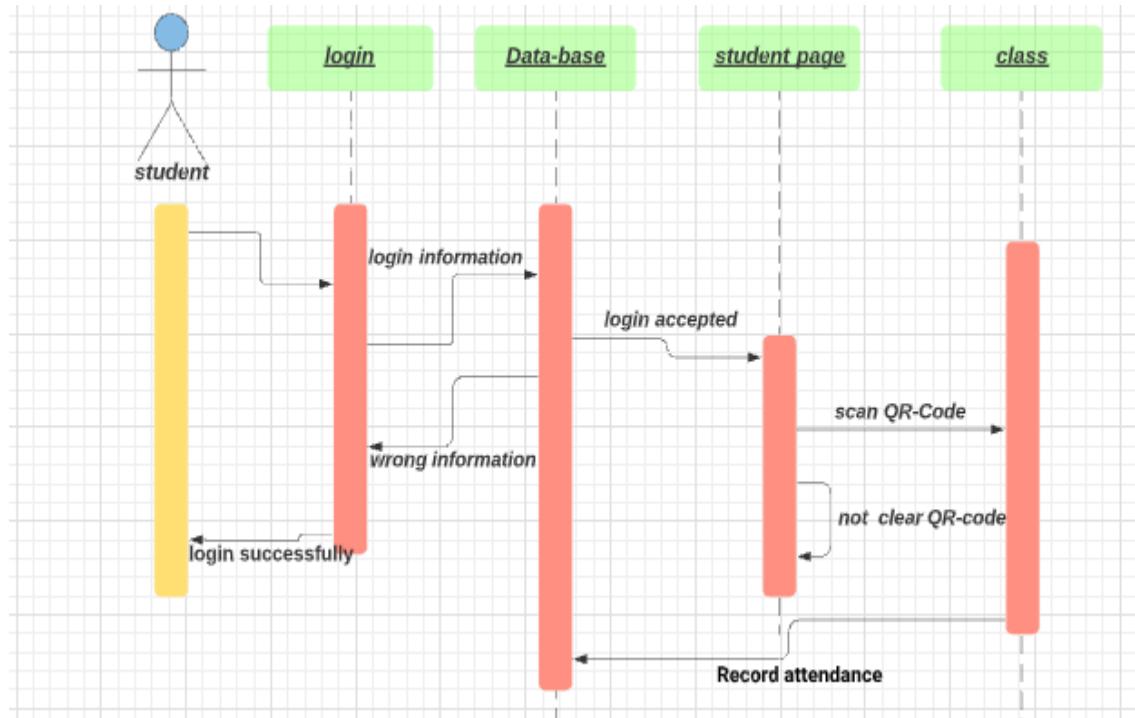


Fig. 2.4: Attendance system sequence diagram

2.2.1.5 Data Flow Model

The data flow diagram of the system is given in figure 2.5. The student and teacher entry system explains how they deal with the data base and some of the teacher's operations such as creating the QR-Code and printing the student attendance report, as well as how the attendance registration process is done automatically in the database and the ability to show the attendance record for the semester.

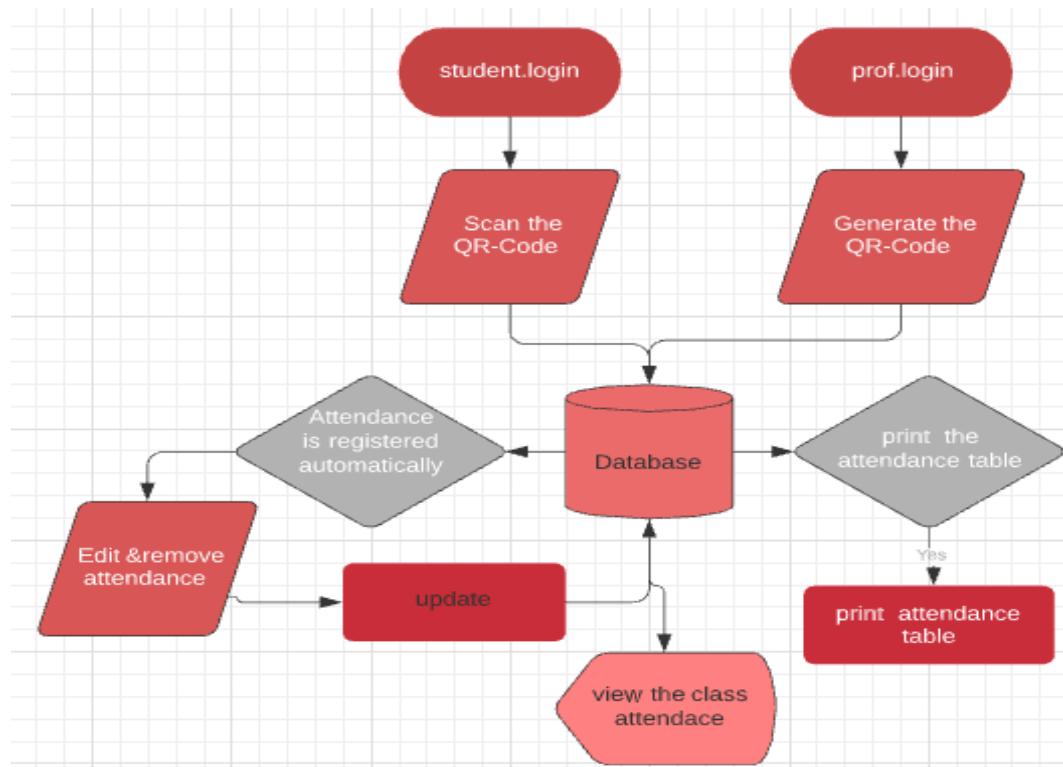


Fig. 2.5: Attendance system data flow diagram

2.2.1.6 The Overall System Structure and System Flowcharts

In this section, the overall structure and the system flowcharts have been displayed. The overall structure is the main parts of the system which cooperated to achieve the main target of the project. The overall structure of the system is shown in figure 6. This parts represents the different platforms used in the projects such as the web system, mobile application, and the system database.

On the other hand, the system flowcharts are represented in figure 2.7 and figure 2.8. These charts displays the instructions of the system operations in both sides “instructor and student”. The system instructions are the steps that each using of the system whether instructor or student are performed to satisfy his needs.

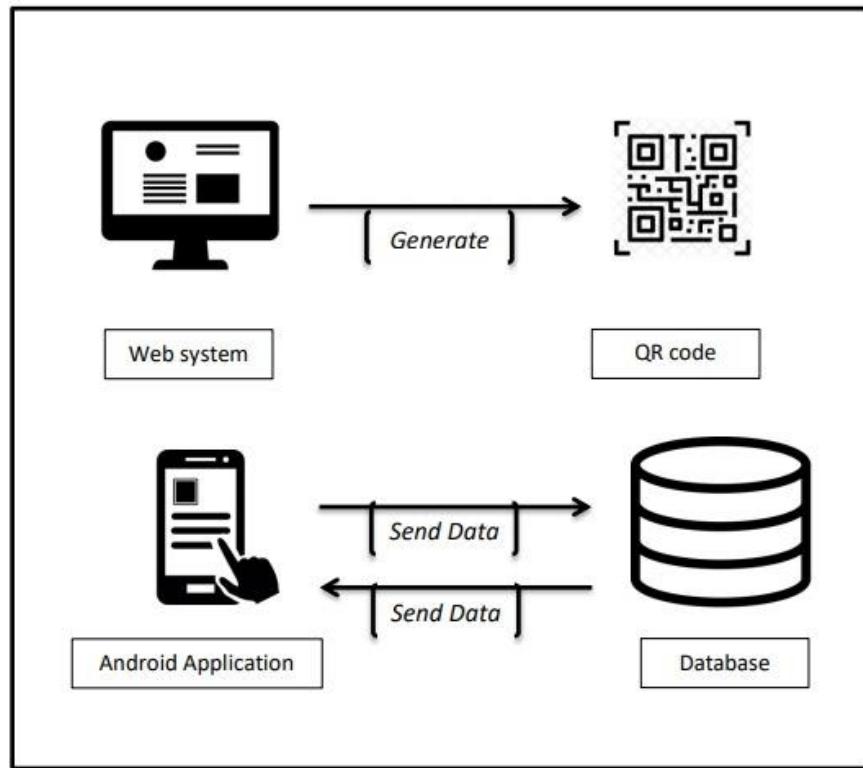


Fig. 2.6: The overall structure of attendance system

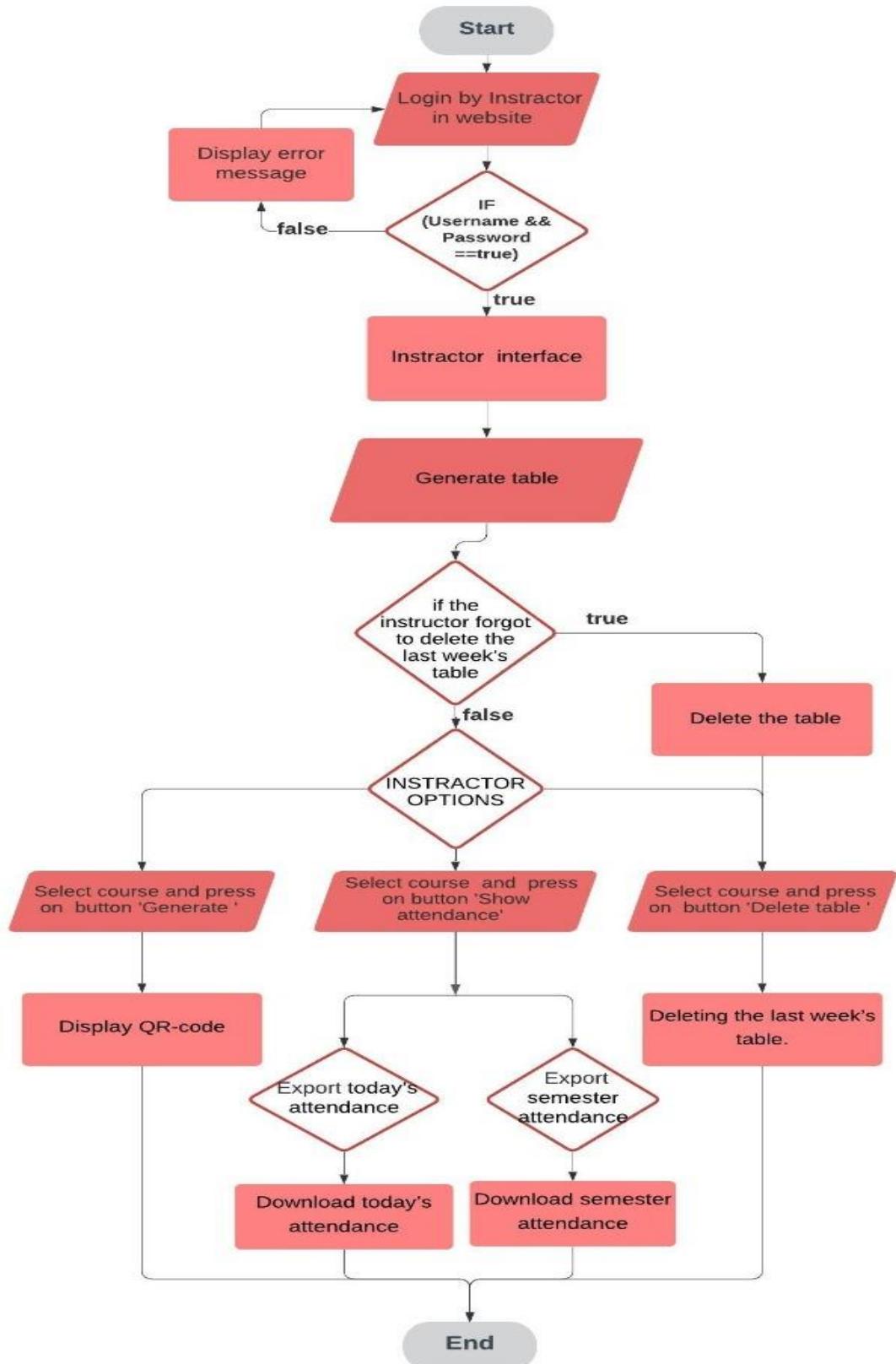


Fig. 2.7: Instructor instructions flowchart

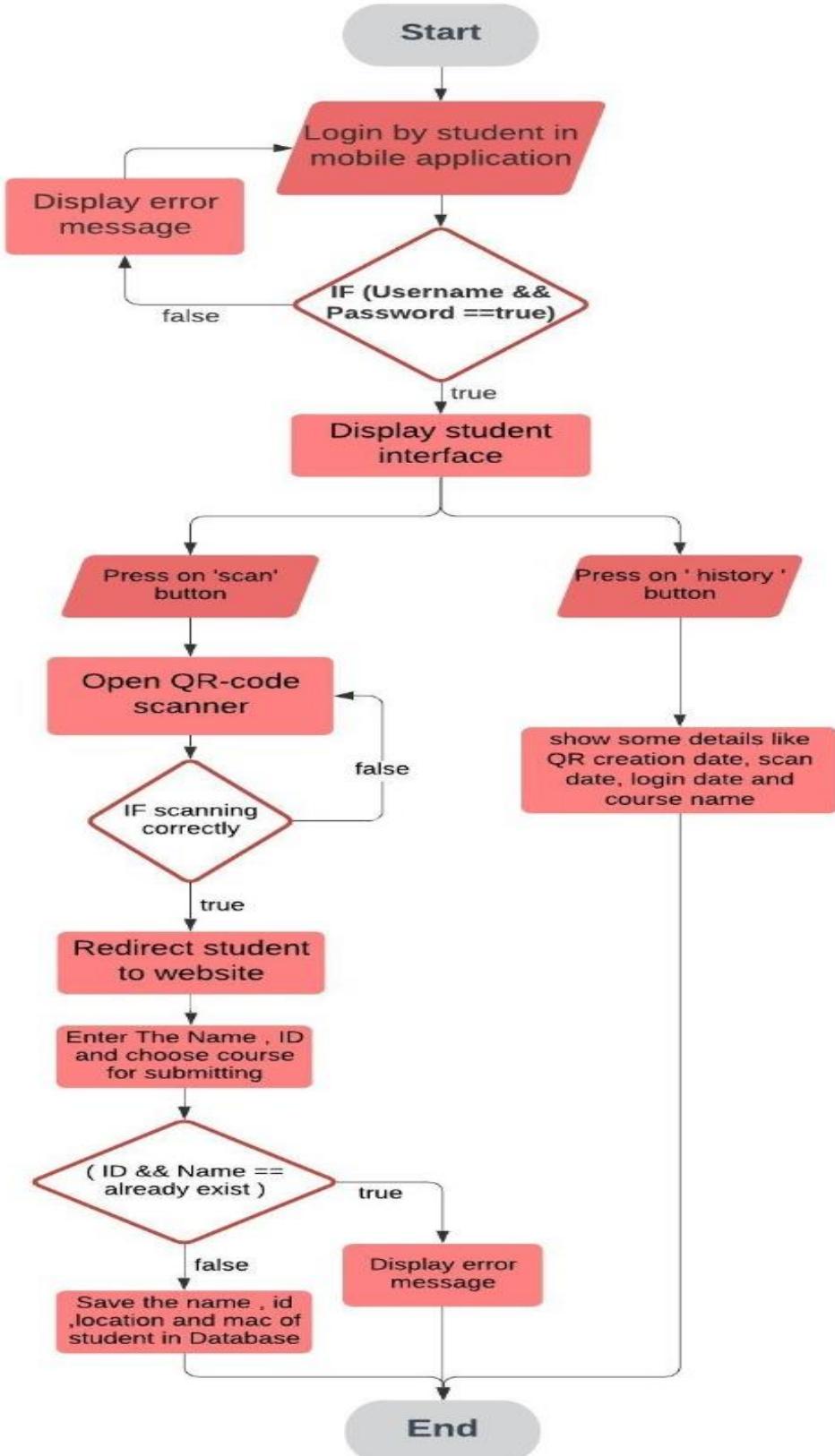


Fig. 2.8: Student instructions flowchart

2.3 Software Platforms

In the proposed attendance system, different software platforms and programming languages have been employed. This variety caused by the variety of the environment which the attendance system dealing with. These platforms are mobile operating systems such as android and IOS to deal with student to scan and acquisition the QR code. Also, PHP, MyPHPAdmin, and java used in a web site development to act as the connection between the instructor process “QR code generation” and the system database. Finally, MongoDB, Firebase, and MySQL are platforms used to construct, organize, and dealing with database and its transactions. In the following sub-section these platforms have been highlighted.

2.3.1 Mobile operating systems

QR codes can be used on various mobile device operating systems. iPhones running on iOS 11 and higher and some Android devices can natively scan QR codes without downloading an external app. The camera app is able to scan and display the kind of QR code (only on iPhone) along with the link (both on Android and iPhone). These devices support URL redirection, which allows QR codes to send metadata to existing applications on the device. Many paid or free apps are available with the ability to scan the codes and hard-link to an external URL.

2.3.2 Website login

QR codes can be used to log into websites: a QR code is shown on the login page on a computer screen, and when a registered user scans it with a verified smartphone, they will automatically be logged in. Authentication is performed by the smartphone which contacts the server. Google tested such a login method in January 2012.

2.3.3 Database

2.3.3.1 MongoDB

MongoDB is an open-source document-stored NoSQL database. It is well integrated with popular programming languages. It is used due to the performance and flexibility of the database structure as shown in figure below:

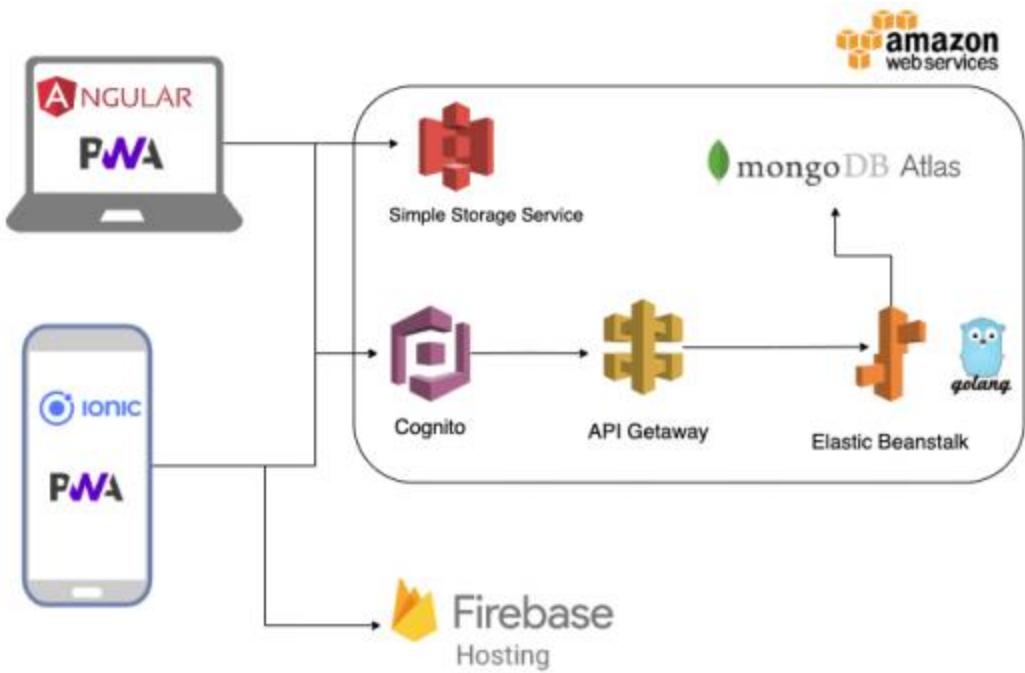


Fig. 2.9: Proposed Solution Architecture Design

2.3.3.2 Firebase

The Firebase Real-time Database is a cloud-hosted NoSQL database that lets you store and sync data between your users in Real-time.

2.3.3.3 PhpMyAdmin

It is an open-source tool and also, it is free written in PHP, XHTML, CSS, and JavaScript planned to manage the administration of MySQL by using of a web. It is able to perform various missions like creating, modifying databases, tables, fields, executing SQL statements or managing and supervise users. PhpMyAdmin is being translated into 72 languages in order to make the usage easy to a wide domain of people and it supports both LTR and RTL languages.

2.3.3.4 PHP

It stands for PHP: Hypertext Preprocessor but, originally stood for Personal Home Page. Is a server-side scripting language that designed for web development, as well as used for general purpose language? It was created in 1994 by Rasmus Lerdorf, in the present time the reference execution of PHP is produced by the PHP group. In January 2013, PHP was installed on more than 240 million websites, and 2.1 million web servers. The PHP code can be combined

with several web frameworks and templating engines or simply it can be mixed with HTML code.

2.3.3.5 MySQL

SQL stands for Structured Query Language. MySQL is an open-source Relational Database Management System (RDBMS); it is a popular database for use in web applications, and is a central part of the greatly used LAMP (Linux, Apache, MySQL, Perl/PHP/Python) open-source web application software stack. MySQL is used by many applications like, WordPress, Joomla, TYPO3, Drupal, MyBB, phpBB, MODX and other software. Numerous large scale websites including Google, YouTube, Facebook, Twitter, and Flickr are also using MySQL.

2.4 Database in our project

First, we going to talk about the database which is an important part of QR-code attendance system because it records all the data taken form students and university, so we chose Xampp software which is using PHP and MySQL language to set up the database, however Xampp is an open-source web development solution stack package that we can use not just in database but in web too later on. Moreover, Xampp is easy to install and use we going to see examples on how to make a database or how to look like in Xampp.

First creating database, you write the name you want for the database in the text box labeled “Database name” then click create as shown in the following:

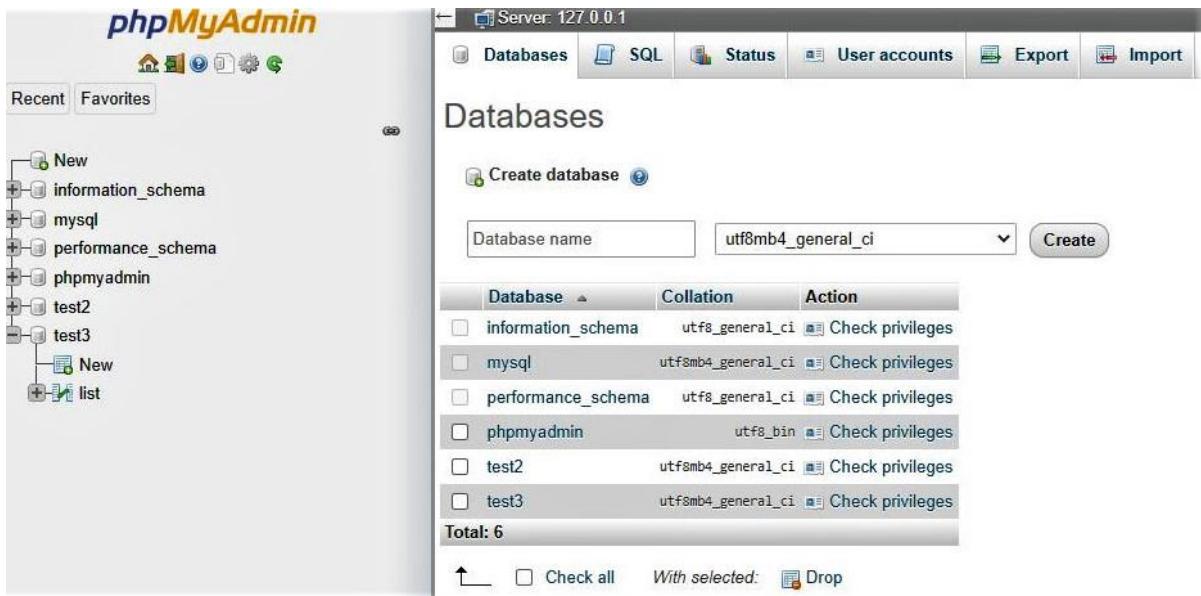


Fig. 2.10: Creating database using XAMPP

Then you create a table write the name you want in text box labeled “Name” and choose the number of columns you want in your table usually in our project it’s the student name and id as shown in the following:

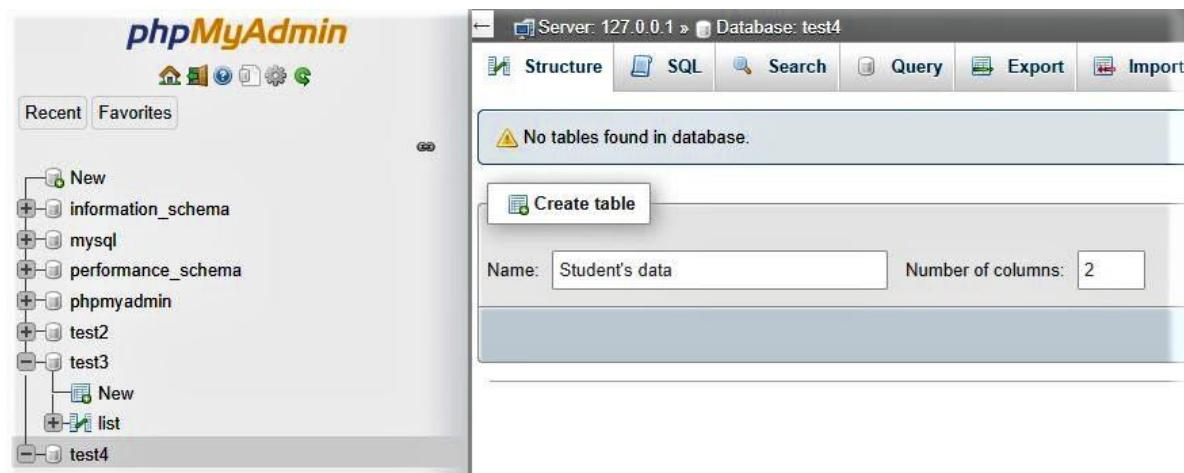


Fig. 2.11: Creating a table in the database

Furthermore, we choose the characteristic of the date that will be registered in the database for student name we choose VARCHAR and INT for student id as shown in the following:

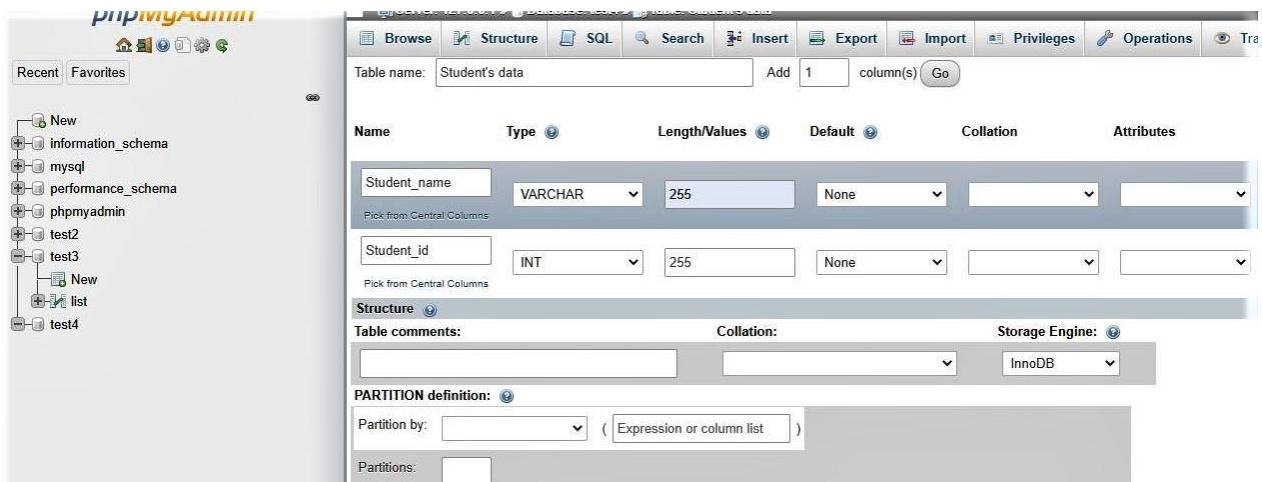


Fig. 2.12: Assigning the characteristics of the table

Then it should look like shown in the following::

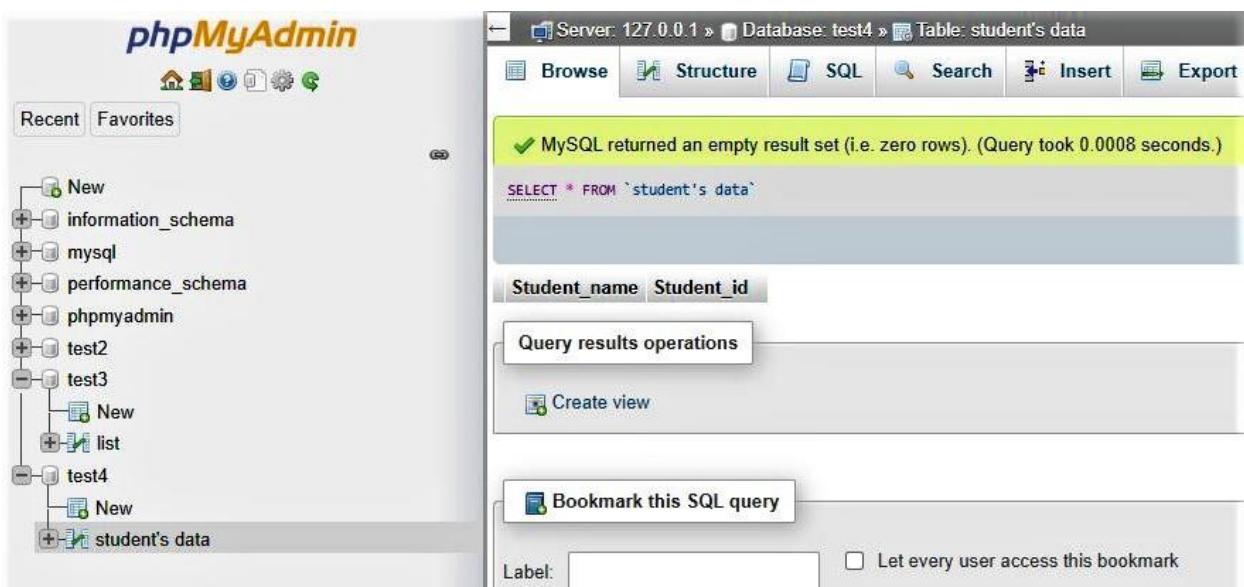


Fig. 2.13: Table shape in database

We can also add an excel sheet of the student's data into the database, we click import then chose the excel file, however the excel file type should be .CSV to accepted in the database because it's easier for web development and it has separated values file, which allows data to be saved in a tabular format as shown in the following:

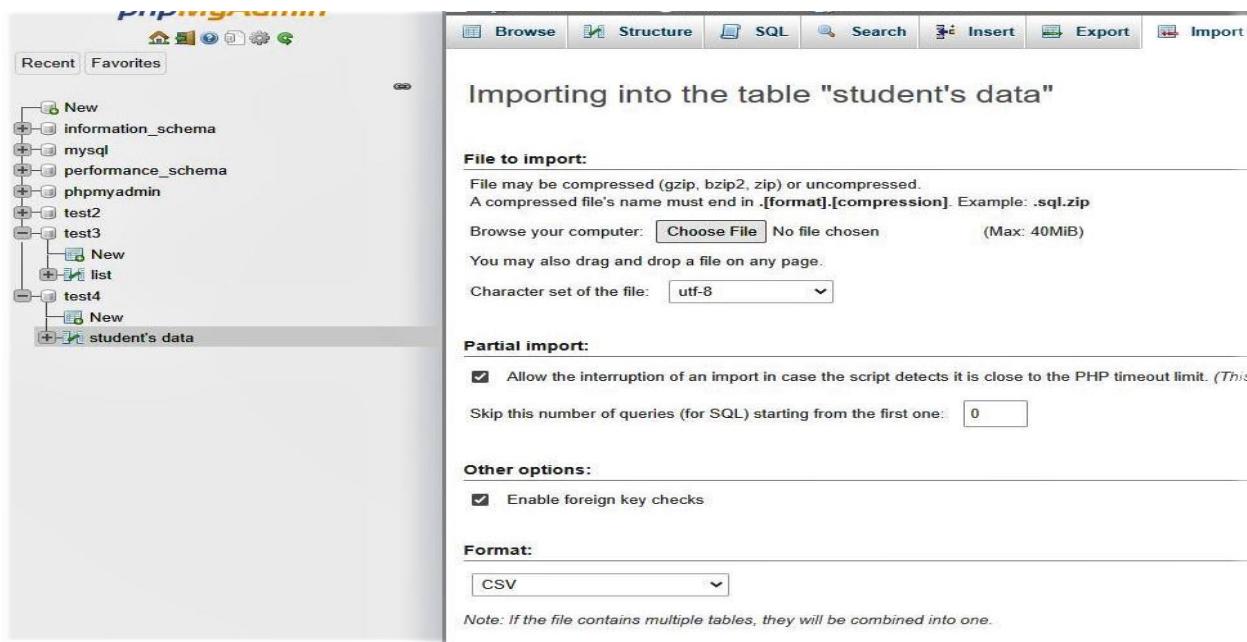


Fig. 2.14: Uploading excel sheet to database

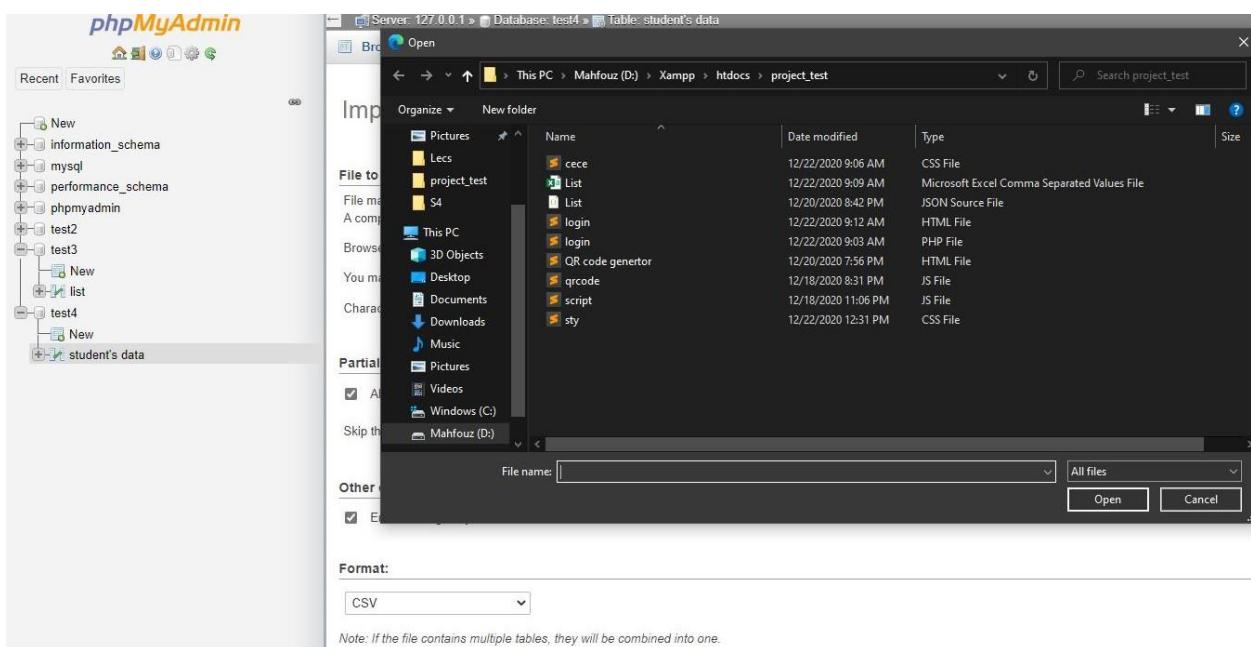


Fig. 2.15: Uploading excel sheet to database

And it should look like as shown in the following::

The screenshot shows the phpMyAdmin interface with the following details:

- Server:** 127.0.0.1 » **Database:** test4 » **Table:** student's data
- Toolbar:** Browse, Structure, SQL, Search, Insert, Export, Import
- Message Bar:** Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete.
- Query Result:** Showing rows 0 - 4 (5 total, Query took 0.0009 seconds.)
- SQL Query:** SELECT * FROM `student's data`
- Table Data:**

Student_name	Student_id
Mahmoud Ashraf Telbany	51610295
Mohamed Mostafa	51610240
Andrew Atef	51610451
Youssef Ahmed	51610425
Mahfouz Assem Mohamed	51610491
- Controls:** Show all, Number of rows: 25, Filter rows: Search this table

Fig. 2.16: How the data in the database looks like

Yet, that was a small example of how we going to use Xampp to create and build a database for the system, additionally we going to need a website which the instructor or the Lecter is going to use in the lecture to generate the QR code to the students for this we will be trying to Html and JavaScript along with Xampp of course.

2.5 The website in our project

In this section we are going to discuss how and what will we use to create our website in the QR code system, although to create a website we will have to use HTML, CSS, and JavaScript along with PHP of course to connect it to the database.

First HTML is the standard markup language for creating Web pages as shown in the following:

The screenshot shows the Visual Studio Code interface with the QRCode.html file open in the editor. The code is as follows:

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="style.css">
    <title>QR Code Generator</title>
</head>
<body>
    <div class="container">
        <form id="generator">
            <!-- Full Name -->
            <input type="text" name="fullName" id="fullName" placeholder="Full Name" required>
            <!-- Sign in button -->
            <button type="submit">Generate</button>
        </form>
        <div id="code" class="code"></div>
    </div>
    <script src="qrcode.js"></script>
    <script src="script.js"></script>
</body>
</html>
```

Fig. 2.17: HTML code example

Second, we are going to use CSS to style web page and this is not the final shape, it's a draft as shown in the following:

The screenshot shows the Visual Studio Code interface with the style_1.css file open in the editor. The code is as follows:

```
.container{
    display: flex;
    flex-direction: column;
    justify-content: space-between;
    margin: auto;
    height: 320px;
    width: 250px;
}

/* qrcode */
#qrcode {
    border: 1px solid black;
    height: auto;
}

.code img{width:100%;height:auto;}
```

Fig. 2.18: CSS code example

Third we are going to use a programming language JavaScript to make our web page dynamic, so we are going to use function to generate dynamic QR code for scan by student mobile to attend in lecture as shown in the following:

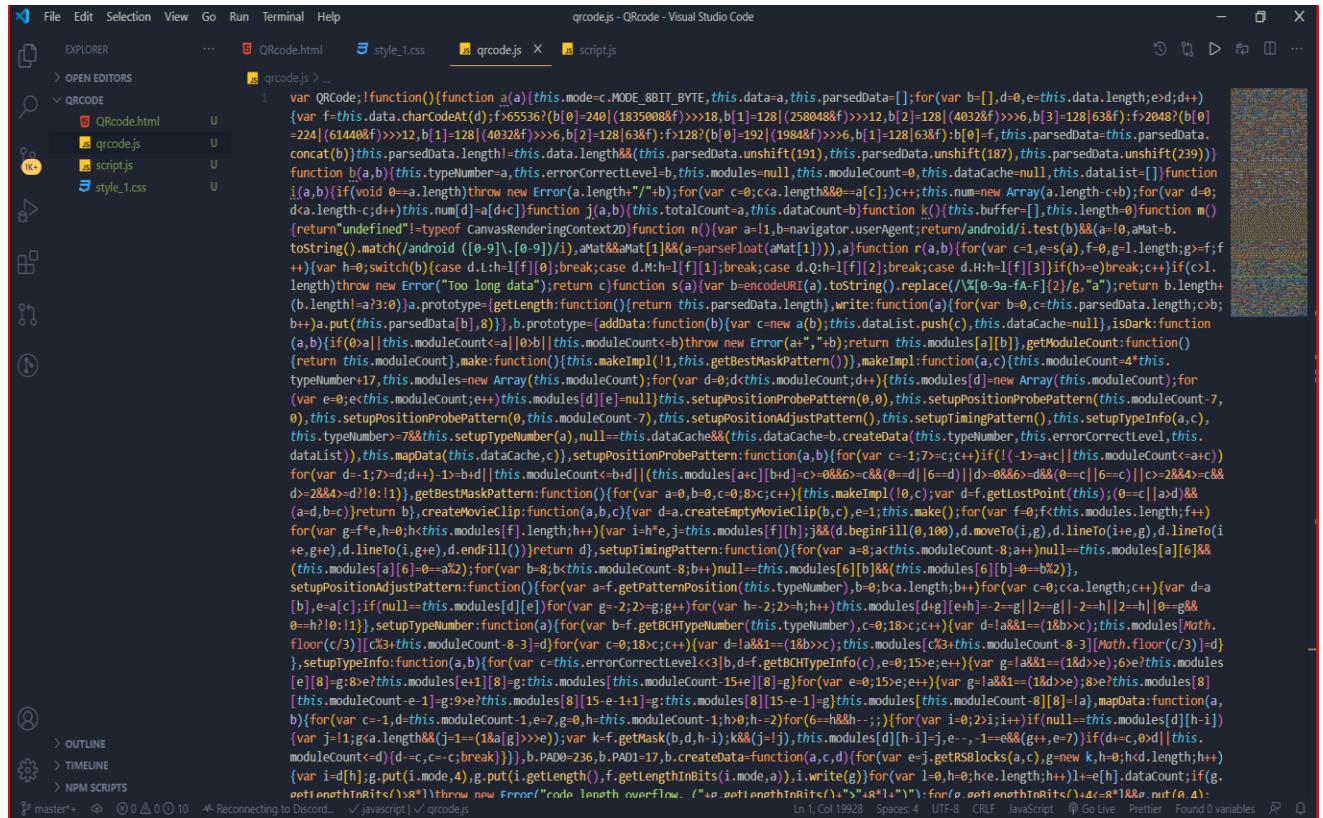


Fig. 2.19: JavaScript QR code example

We still using JavaScript to generate QR code by adding some data to get out QR code in high quality to make it apparent, on another hand we need to secure our processes to prevent cheating in attendance, so we think about how to solve this problem and get out more steps to try making system secure by using geolocation and we are adding JSON to collect name from input, date and location as shown in the following:

The screenshot shows a Visual Studio Code interface with a dark theme. The top menu bar includes File, Edit, Selection, View, Go, Run, Terminal, and Help. The left sidebar has sections for Explorer, QRCode, Outline, Timeline, and NPM Scripts. The main editor area contains a file named 'script.js' with the following code:

```
script.js - QRcode - Visual Studio Code

script.js > getLocation > <function> > success
  1 const form = document.getElementById('generator')
  2 const fName = document.getElementById('fullName')
  3
  4 let current = {}
  5
  6 getLocation().then(function(result) {
  7   current = result
  8 })
  9
 10 form.addEventListener('submit', function(e) {
 11   e.preventDefault();
 12
 13   this.style.display = 'none';
 14
 15   new QRCode('code', {
 16     text: JSON.stringify({
 17       name: fName.value,
 18       date: new Date(),
 19       location: current
 20     }),
 21     width: 400,
 22     height: 400,
 23     colorDark: "#000000",
 24     colorLight: "#ffffff",
 25     correctLevel: QRCode.CorrectLevel.H
 26   })
 27 }, false)
 28
 29 function getLocation() {
 30   return new Promise(function(resolve, reject) {
 31     if (!navigator.geolocation) {
 32       reject('Geolocation is not supported by your browser');
 33       return;
 34     }
 35
 36     function success(position) {
 37       var latitude = position.coords.latitude;
 38       var longitude = position.coords.longitude;
 39     }
 40   })
 41 }
 42
 43
 44
 45
 46
 47
 48
 49
 50
 51
 52
 53
 54
 55
 56
 57
 58
 59
 60
 61
 62
 63
 64
 65
 66
 67
 68
 69
 70
 71
 72
 73
 74
 75
 76
 77
 78
```

The status bar at the bottom shows 'Ln 43, Col 9' and other file-related information.

Fig. 2.20: JavaScript location code example

We create new function we called it get location to activate navigator in our browser and return out latitude and longitude of the device when generate QR code, so this process will make it more secure and not easy for cheating because of latitude and longitude will let us know who generate QR code in different places which will not be same position of university and another step we used date and time to make it difficult to guess seconds of generating QR code as shown in the following:

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Editor:** The main area displays the `script.js` file content. The code is a JavaScript function that uses the Geolocation API to get the user's location and then generates a QR code from that information.
- Explorer:** On the left, the Explorer sidebar shows a project structure with files: `QRcode.html`, `style.css`, `qrcode.js`, and `script.js`.
- Bottom Status Bar:** Shows the file path as `script.js - QRcode - Visual Studio Code`, line count as `Ln 43, Col 9`, and encoding as `UTF-8`.

```
function getLocation() {
    return new Promise(function(resolve, reject) {
        if (!navigator.geolocation){
            reject('Geolocation is not supported by your browser');
        }
        resolve([
            latitude,
            longitude
        ])
    })
}

function success(position) {
    var latitude = position.coords.latitude;
    var longitude = position.coords.longitude;

    navigator.geolocation.getCurrentPosition(success, error);
}
```

Fig. 2.21: JavaScript code example

Now, all data will be saved in database to record every student attendance in any lecture to get his right marks without cheating. And this is the frontend of the QR generator where the professor enters the name of the course and generate the QR code as shown in the following:

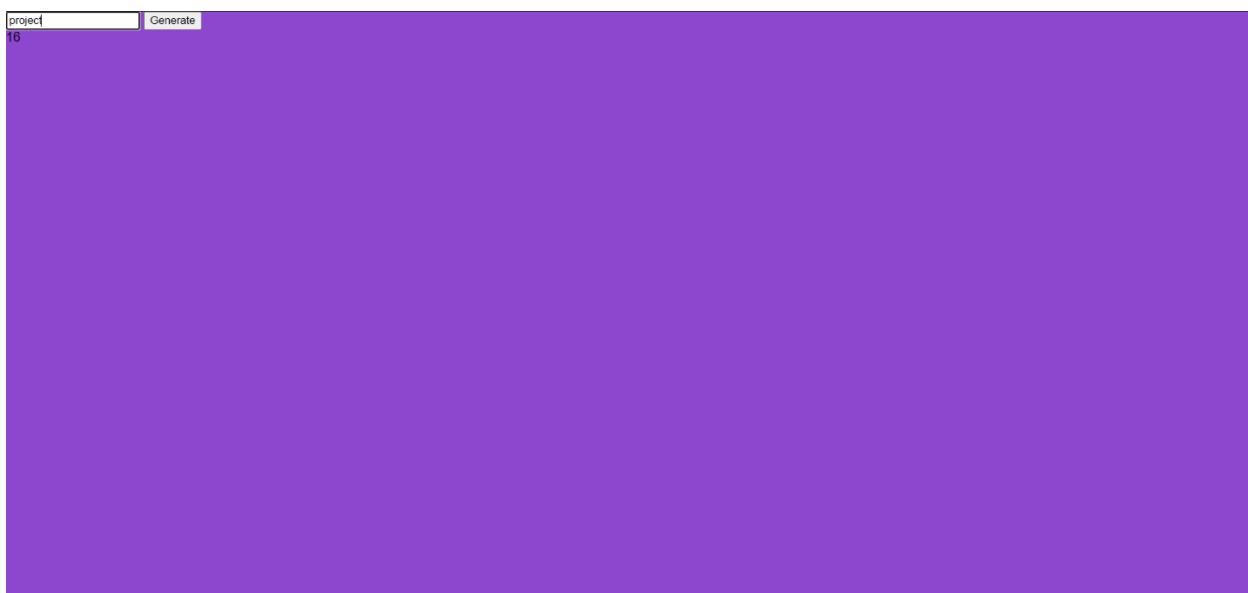


Fig. 2.22: Fronted QR code generator

As shown in the figure below the name of the course, date, longitude and latitude are saved in the QR code for the security as shown in the following:

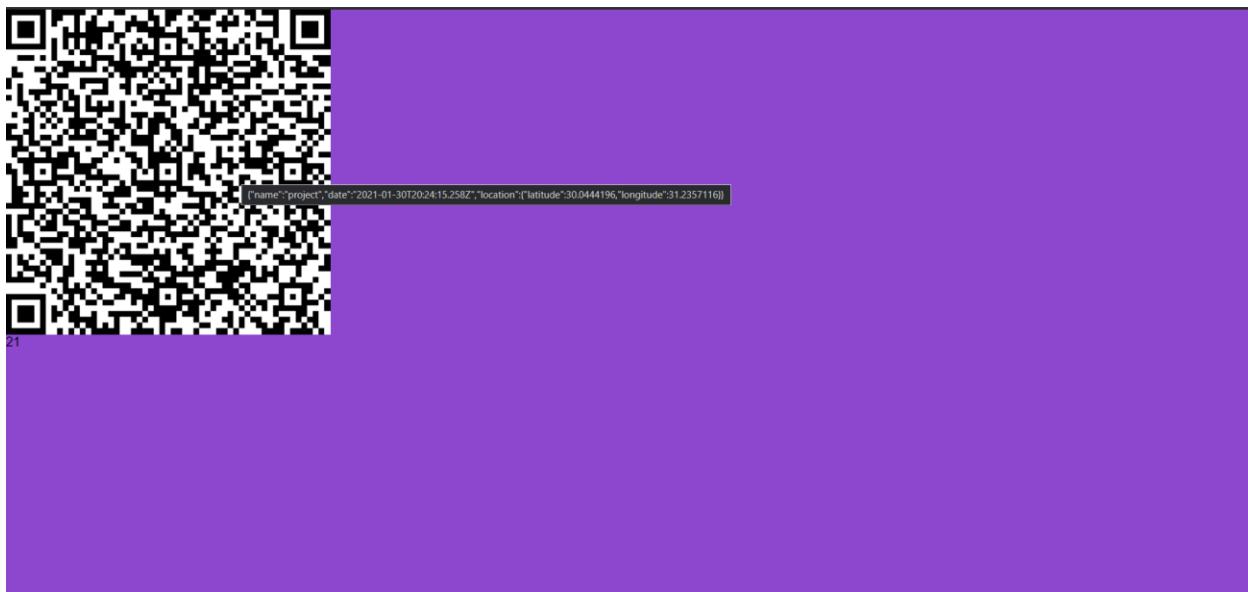


Fig. 2.23: Data represented from the QR code

2.6 COVID-19 pandemic challenging and motivation

After the COVID-19 pandemic began spreading, QR codes began to be used as a "touchless" system to display information, show menus, or provide updated consumer information, especially in the hospitality industry. Restaurants replaced paper or laminated plastic menus with QR code decals on the table, which opened an online version of the menu. This prevented the need to dispose of single-use paper menus, or institute cleaning and sanitizing procedures for permanent menus after each use. In several Australian state's patrons were required to scan QR codes at shops, clubs, supermarkets and other service and retail establishments on entry to assist contact tracing. Singapore, the United Kingdom and New Zealand used similar systems

2.7 Future Work

By the ending of part one work of the proposed attendance system, we are going to improve the UI of the system to fit todays perspective in both website and the mobile application. In addition, the security will be enhanced to the maximum level and solve all the tricky and common issues of the QR CODE attendance. furthermore, improve its efficiency to make it easier and faster to the user. We will work on the front-end by using a programming language like JavaScript, styling by CSS and Scripting by HTML, and for the back-end we will use scripting language like PHP and programing language like SQL. In the next chapter, the part two of the attendance system development and enhancing will be highlighted and discussed in details.

Chapter 3

Practical work and user instructions

Introduction

In this chapter, the forms and user interfaces have been displayed. The user interfaces are the frontage of the attendance system that users whether instructors or students are dealing with. On the other side, the code of each programming platform used in different stages are displayed in this chapter. As discussed in the previous two chapters, employing QR code technology in building an attendance system are a big challenge in different levels such as security level, stability level, maintainability level, and efficiency level. The main goal of this chapter is trying to display and test all the instructions of the system in different levels with different users. The attendance system controlling is divided into two main parts website part and mobile application part. In the next sections, the chapter will highlight the different steps and instructions within these two parts.

3.1 Website Part

The website part is the main core of the attendance system. It consists of the login page of the instructor, student submitting page, table or course selection, generating QR-code, showing the attendance table, export, and download the attendance report. The other part “mobile application” is acting as a QR code scanner application performed by students.

3.1.1 QR-Code Generation

Now we will see how the website functions are performed efficiently with screen shots. Firstly, in the beginning of using the website of attendance system it directly login to the instructor home page as shown in the figure 3.1.

This home page is used by every instructor who has permission to login to the website and use the QR-code attendance system. Also, we are going to display the source code of creating functioning of each page. The PHP code of creating the instructor database login have been shown in figure 3.2.

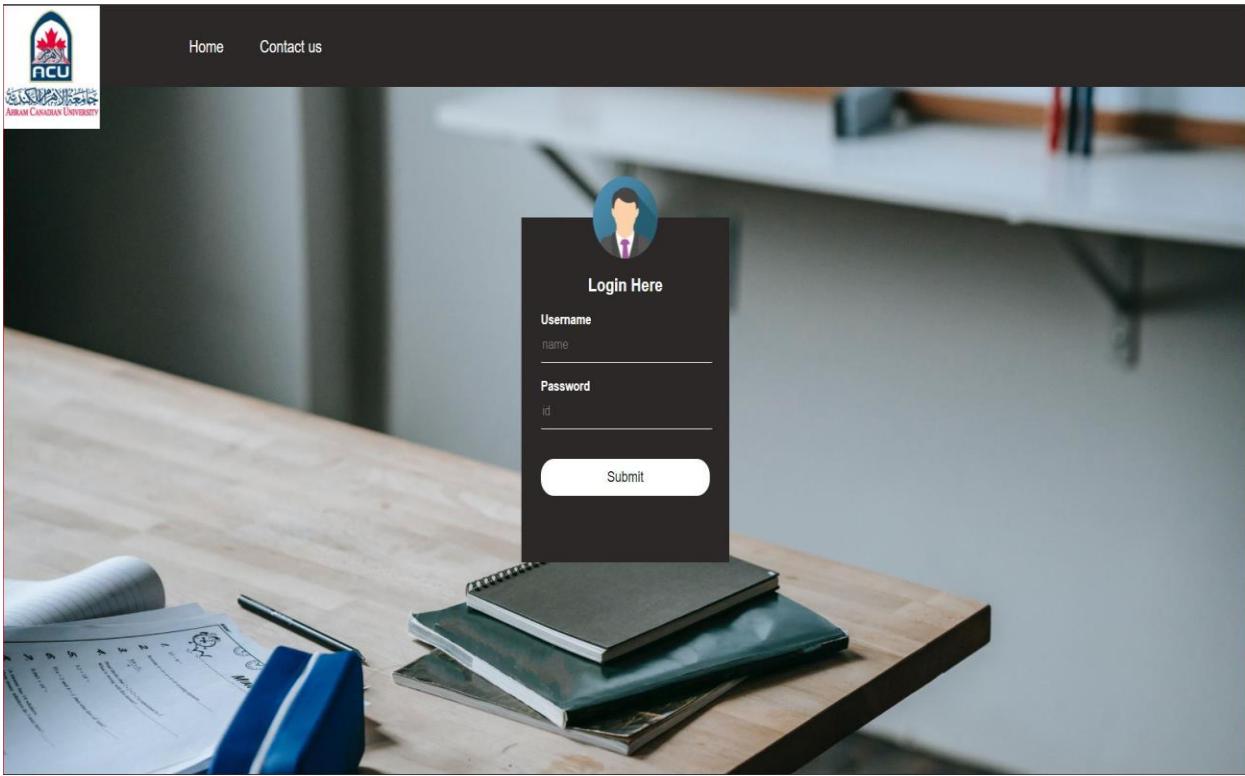


Fig. 3.1: Instructor login home page

```
1  ?>php
2
3  session_start();
4
5  $con=mysqli_connect("localhost","root","","qr")or die("connection failed");
6
7  mysqli_select_db($con,'qr');
8
9  $name=$_POST['name'];
10 $id=$_POST['id'];
11
12 $sql = " SELECT * FROM teacher WHERE name = '$name' &amp;&amp; id = '$id' ";
13
14 $result = mysqli_query($con, $sql);
15 $num = mysqli_num_rows($result);
16
17 if($num==1)
18 {
19     header('location:table.html');
20 }
21 else {
22     header('location:login_database_link.html');
23 }
24
25 ?|</pre
```

Fig. 3.2: Instructor PHP code for login function

After the instructor successfully login, the instructor will be redirected to the table generating page or course selection page where he chooses the course for today's attendance as shown below:

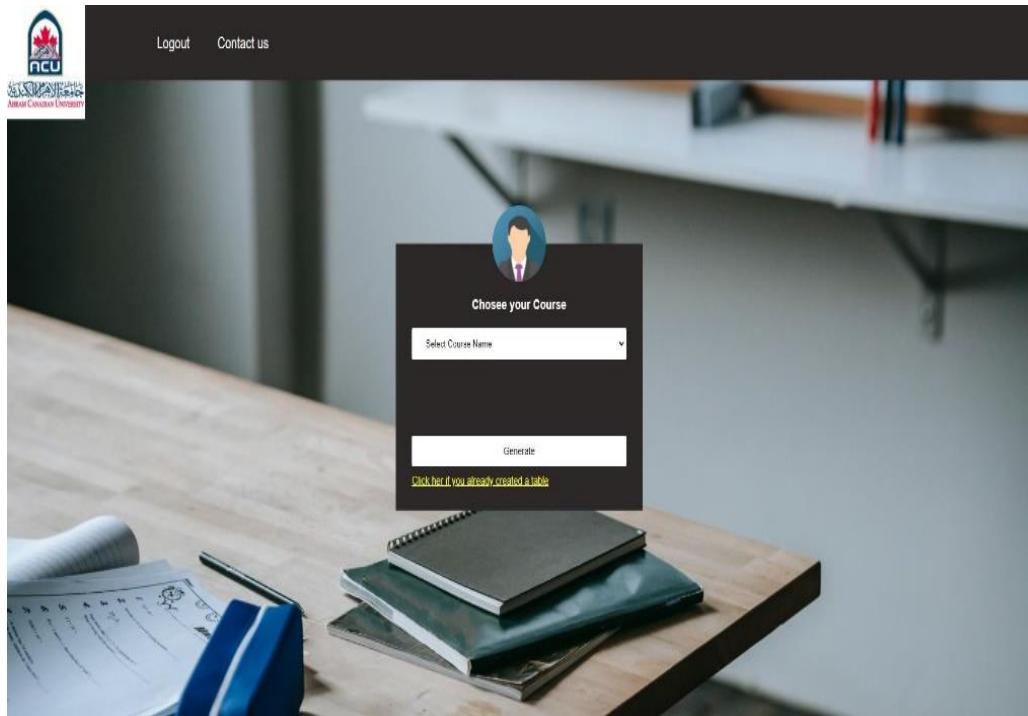


Fig. 3.3: Current lecture attendance creation page

The instructor choosing the course from the list of courses available in the system and stored before in the system database through the university management. After course selection, the instructor clicks generate button which will generate the table of the course that chosen. For the first use of attendance system the table of current lecture attendance is empty and no need for deleting the content of that table. After current lecture attendance process completed, the attendance is stored directly to the overall semester attendance. In the next use of the current lecture attendance table the lecture has to ensure that it empty. This step can be achieved by deleting current lecture attendance content in the end of last session or in beginning of the new session before the new lecture attendance process begins. However, if the instructor forgot to delete the last week's table, then the instructor can click down below to move to the next page to delete the last lecture attendance table, also when you move to the other page you will be notified to delete last week's table with an alert message in case the instructor forgot as shown in the following:

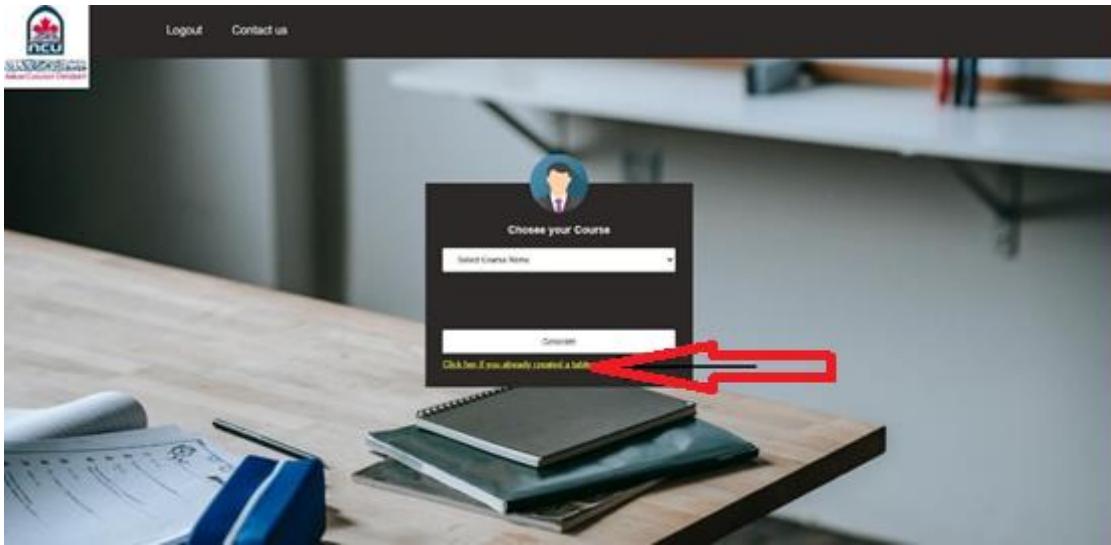


Fig. 3.4: Forgetting to delete last week attendance

On the other hand, the code used for the course creation function show below:

```
1 <?php
2     // Create connection
3     $con=mysqli_connect("localhost","root","","qr")or die("connection failed");
4     $fullName=$_POST['fullName'];
5
6     // sql to create table
7     $sql = "CREATE TABLE `{$fullName}` (
8         `name` varchar(50) NOT NULL,
9         `id` varchar(50) NOT NULL,
10        `ip` varchar(50) NOT NULL Primary key,
11        `date` datetime(6) NOT NULL
12    ) ";
13
14
15    if ($con->query($sql) === TRUE) {
16
17        header('location:QRcode.html');
18
19    } else {
20        echo "Error creating table: " . $con->error;
21    }
22
23    $con->close()
24 ?>
```

Fig. 3.5: Table creation PHP code

Now we can see current lecture attendance table in the database, also we can see the tables of the overall semester attendance where the attendance data for all students are stored cumulatively. The current lecture attendance table and overall semester attendance tables in database are shown in figure 3.6.

Table	Action	Rows	Type	Collation	Size	Overhead
image	[Browse](#) [Structure](#) [Search](#) [Insert](#) [Empty](#) [Drop](#)	1	InnoDB	utf8mb4_general_ci	16.0 Kib	-
image_count	[Browse](#) [Structure](#) [Search](#) [Insert](#) [Edit](#) [Drop](#) [View](#)	~8	InnoDB	utf8mb4_general_ci	-	-
image_total_attendance	[Browse](#) [Structure](#) [Search](#) [Insert](#) [Empty](#) [Drop](#)	2	InnoDB	utf8mb4_general_ci	16.0 Kib	-
teacher	[Browse](#) [Structure](#) [Search](#) [Insert](#) [Empty](#) [Drop](#)	3	InnoDB	utf8mb4_general_ci	16.0 Kib	-
4 tables	[Sum](#)	~6	InnoDB	utf8mb4_general_ci	48.0 Kib	0 B

 A 'Create table' form is visible at the bottom of the main area, with 'Name' set to 'New' and 'Number of columns' set to '4'. The 'Go' button is present at the bottom right of the form.

Fig. 3.6: current lecture attendance table and overall semester attendance tables in database

After creating or choosing your desired course, we redirected to the QR-code generation page. Now it can generate the QR-code and show the attendance report as shown in the following figure 7 and figure 8. This page contains in addition to generate QR-code and display the attendance report, there is the button for deleting the last week's attendance table if the instructor missed to delete it in the last session as it discussed before. Also, a button that transfer attendance from table today's attendance to table total attendance has been deployed.

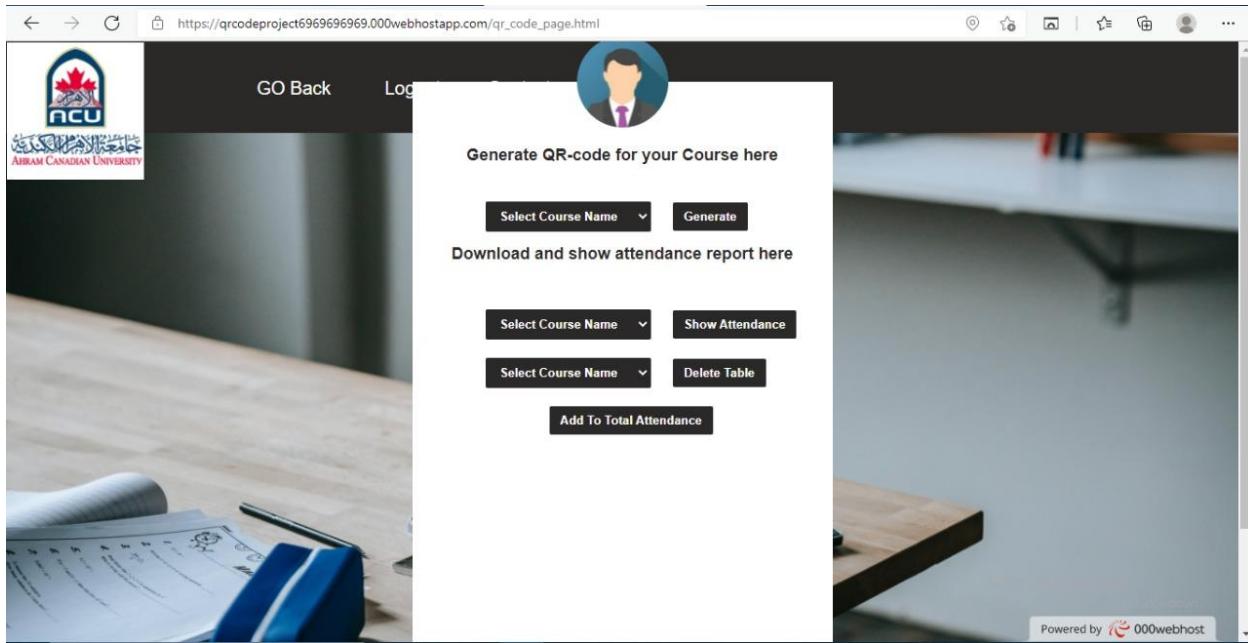


Fig. 3.7: QR-code generating page in addition to different attendance reports

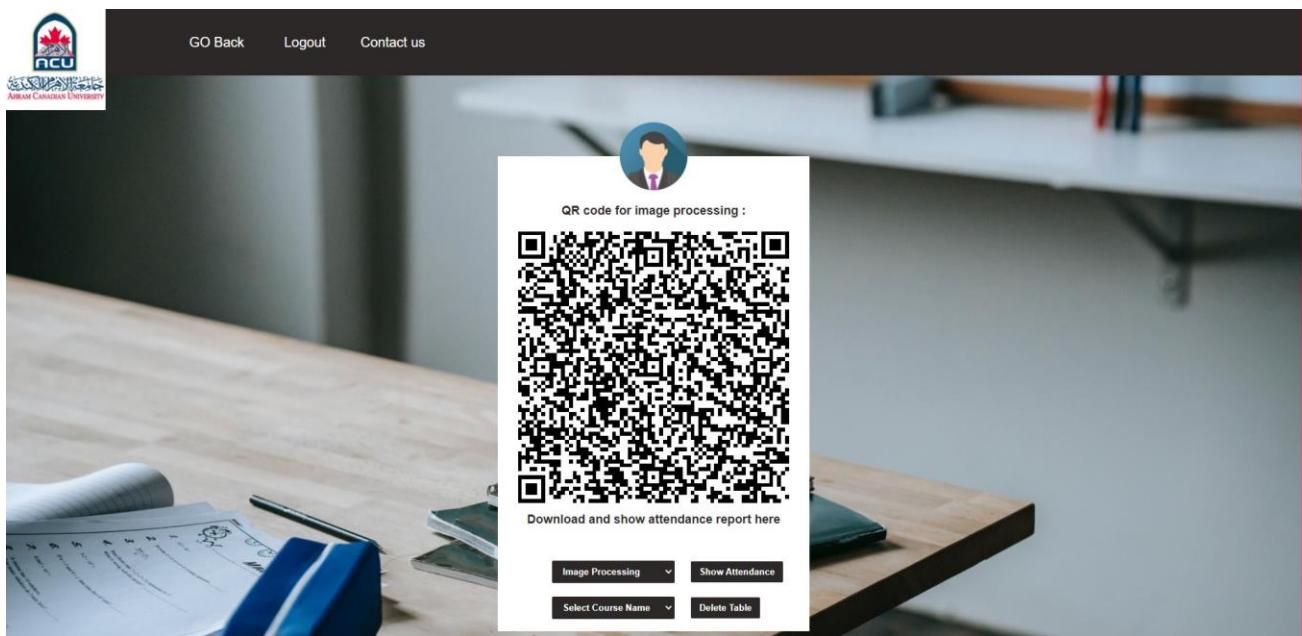


Fig. 3.8: QR-code generating page in case of QR-code generating

After generating QR-code there will be a session time for the students to scan the QR-code after that the page will reset. This time is controlled to be a small time near to 30 seconds to ensure session security and ensure that no any jugglery operations have been performed.

3.1.2 Student Submission

Now, the student will scan the generated QR-code with the mobile application that we will talk about it in details later in section 3.2, as shown in figure 3.9.

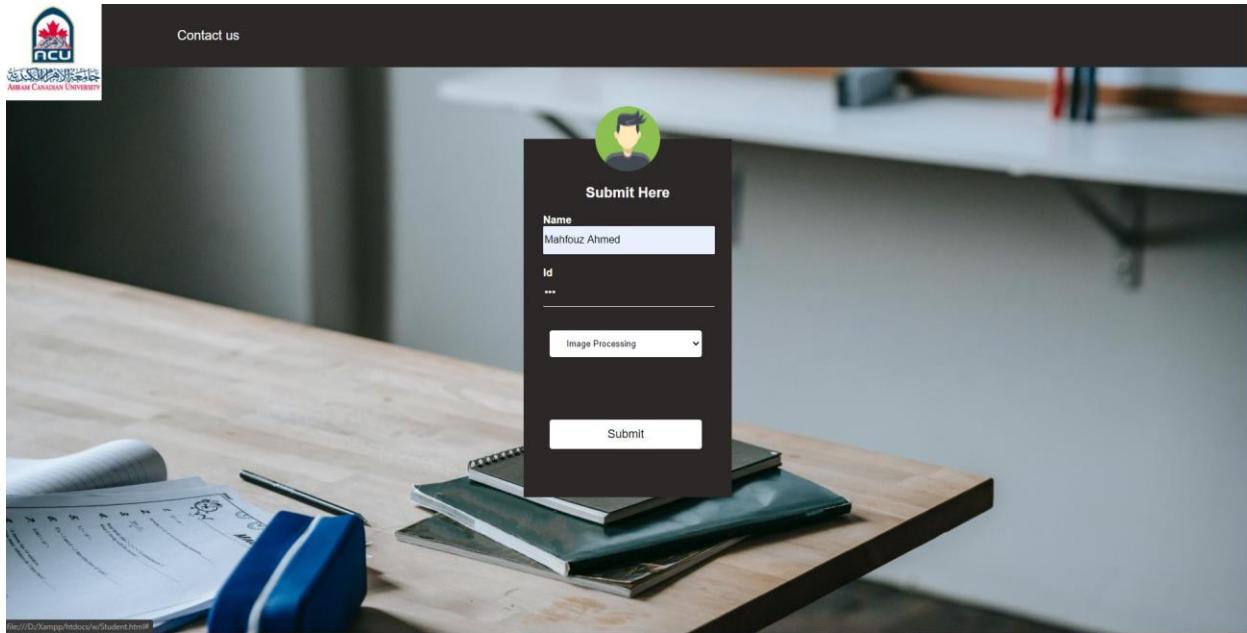


Fig. 9: Student submitting page

The PHP code for creating the student submitting form is shown in figure 3.10. Also, the JavaScript code used for the QR code generating process is consisted of two code pages where one for the QR code itself and the other for the functions of QR code. First the QR-code.js page is displayed in figure 3.11 and secondly, the JavaScript functions code is displayed in figure 3.12.

```

1  <?php
2  session_start();
3
4  $conn = mysqli_connect("localhost","root","","qr");
5  $IP = $_SERVER['REMOTE_ADDR'];
6  date_default_timezone_set('Africa/Cairo');
7
8  mysqli_select_db($conn,'qr');
9  $name=$_POST['name'];
10 $id=$_POST['id'];
11 $fullName=$_POST['fullName'];
12
13 $date = date("Y-m-d H:i:s");
14
15
16 $sql = "SELECT * FROM $fullName WHERE ip = '$IP'";
17 $s = "SELECT * FROM $fullName WHERE id = '$id'";
18 $result= mysqli_query($conn,$sql);
19 $res= mysqli_query($conn,$s);
20 $num= mysqli_num_rows($result);
21 $num1= mysqli_num_rows($res);
22
23 if($num||$num1==1){
24     header('location:ss.html');
25 }
26
27
28 else{
29     $reg=" INSERT INTO $fullName (name, id, ip, date) VALUES('$name','$id','$IP','$date')";
30     mysqli_query($conn,$reg);
31     header('location:sss.html');
32 }
33 ?>

```

Fig. 3.10: Student submitting page creation PHP code

Fig. 3.11: QR-code.js page

```
1 const form = document.getElementById('generator')
2 const formName = document.getElementById('fullName')
3 |
4 let current = {}
5
6 getLocation().then(function(result) {
7     current = result
8 })
9
10 form.addEventListener('submit', function(e) {
11     e.preventDefault();
12
13     var qrtext = document.getElementById("fullName").value;
14     document.getElementById("qr-result").innerHTML = "QR code for " + qrtext + " :";
15
16     this.style.display = 'none'
17
18     new QRCode('code', {
19         text: JSON.stringify({
20             src: "https://qrsystem2021.000webhostapp.com/Student.html",
21             name: formName.value,
22             date: new Date(),
23             location: current
24         }),
25         width: 400,
26         height: 400,
27         colorDark: "#000000",
28         colorLight: "#ffffff",
29         correctLevel: QRCode.CorrectLevel.H
30     })
31 }, false)
32
33 function getLocation() {
34     return new Promise(function(resolve, reject) {
35         if (!navigator.geolocation) {
36             reject('Geolocation is not supported by your browser');
37             return;
38         }
39
40         function success(position) {
41             var latitude = position.coords.latitude;
42             var longitude = position.coords.longitude;
43
44             resolve({
45                 latitude,
46                 longitude
47             })
48         }
49     })
50 }
51
52 
```

Fig. 3.12: JavaScript functions

Due to the system attendance design, the IP address is employed in the registering process to ensure that each student cannot register for any other students only register for him and for one time. if a student finish his registry operations and try to register attendance for another student the system is denying this process. Now if someone tried to submit an attendance twice from the same device, he/she will not be registered and will get the “You Have Already Registered” message as shown in figure 3.13.

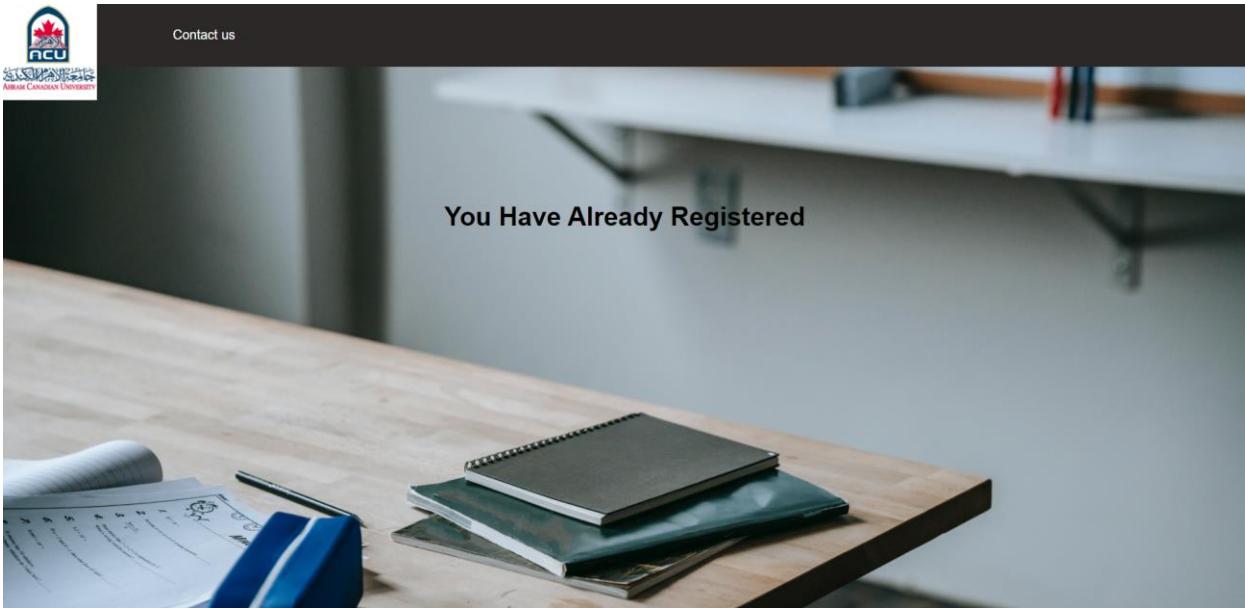


Fig. 3.13: Attendance denying message for twice registry from the same mobile

Here is the table of the course that student's attendance has been stored. As shown in the figure 3.14, the students are registered with their name, id, IP address, and date of the lecturer.

	name	id	ip	date
	Mahfouz Ahmed	51610491	2021-05-23 23:25:56	0.000000

Fig. 3.14: Preview of the lecture attendance table in the database

The instructor can click on show attendance button as shown in page in the figure 3.7 to show today's attendance report where the attended students are listed. Also, current lecture attendance or the overall semester attendance can be downloaded in an excel report as shown in figure 3.15.

Name :	Id :	Date :
Mahfouz Ahmed	51610491	2021-05-06 23:13:18.000000
Mahmoud Ashraf Telbany	51610295	2021-05-06 23:13:18.000000
Andrew Atef	51610451	2021-05-06 23:13:18.000000
Youssef Ahmed	51610425	2021-05-06 23:13:18.000000
Mohamed Mostafa	51610240	2021-05-06 23:13:18.000000

Select Course Name Select Course Name

Fig. 3.15: Students' attendance reports

For the overall semester attendance there is a table which responsible for the semester attendance as shown in figure 3.16. There is another option to the instructor to use a button which responsible for counts the total attendance of the student. This option is designed in the attendance system by using a view script to count each student attendance number from overall semester attendance. Also, another option to download the report that contains count number of attendance for all students as shown in figure 3.17.

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 1 (2 total). Query took 0.0000 seconds.

```
SELECT * FROM `image_total_attendance`
```

	name	id	ip	date
1	Mahfouz Ahmed	51610491	10.0.3.11	2021-07-01 10:03:11.000000
2	Mahfouz Ahmed	51610491	10.0.4.13	2021-07-01 10:04:13.000000

Fig. 16: Semester total attendance table

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 0 (1 total). Query took 0.0011 seconds.

```
SELECT * FROM `image_count`
```

	id	name	COUNT(id)
1	51610491	Mahfouz Ahmed	2

Fig. 17: Semester view table attendance counter

As displayed in the table each student has student's name, id, and scanned date. The HTML/PHP code used for attendance table creation in the website is displayed in figure 3.18.

```

1  <?php
2  $connect = mysqli_connect("localhost", "root", "", "qr");
3  $fullName=$_POST['fullName'];
4  $sql = "SELECT * FROM $fullName";
5  $result = mysqli_query($connect, $sql);
6  ?>
7  <html>
8  <head>
9   <title>Attendance Table</title>
10 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css"/>
11 <link rel="stylesheet" href="exportcss.CSS"/>
12 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>
13 <script src="https://ajax.googleapis.com/ajax/libs/jquery/2.2.0/jquery.min.js"></script>
14 </head>
15 <body>
16
17 <div class="logo">
18 <a href="#"> </a>
19 <nav>
20 <ul>
21 <li><a href="">Contact Us</a></li>
22 </nav>
23 </ul>
24 </div>
25 <div class="container">
26   <br />
27   <br />
28   <br />
29   <div class="table-responsive">
30     <h2 align="center">Attendance Table</h2><br />
31     <table class="table table-bordered">
32       <tr>
33         <th>Name : </th>
34         <th>Id : </th>
35         <th>Date : </th>
36
37       </tr>
38
39     <?php
40     while($row = mysqli_fetch_array($result))
41     {
42       echo '
43       <tr>
44         <td>'.$row["name"].'</td>
45         <td>'.$row["id"].'</td>
46         <td>'.$row["date"].'</td>
47
48       </tr>
49       ';
50     }
51   }
52   ?>
53 </table>
54 </div>

```

Fig. 18: Attendance list HTML/PHP creation code

So, the instructor will download current lecture attendance report or the overall semester attendance report by pressing the buttons export today's attendance or export semester attendance as shown in figure 3.19. The excel.csv file will download and it have the attendance report content.

The screenshot shows a table titled "Attendance Table" with the following data:

Name :	Id :	Date :
Mahfouz Ahmed	51610491	2021-05-06 23:13:18.000000
Mahmoud Ashraf Telbany	51610295	2021-05-06 23:13:18.000000
Andrew Atef	51610451	2021-05-06 23:13:18.000000
Youssef Ahmed	51610425	2021-05-06 23:13:18.000000
Mohamed Mostafa	51610240	2021-05-06 23:13:18.000000

Below the table are two sets of buttons:

- Top row: "Select Course Name" dropdown, "Export today's attendance" button.
- Bottom row: "Select Course Name" dropdown, "Export semester attendance" button.

Fig. 3.19: Downloading different attendance reports

Just by pressing export today's attendance or export semester attendance button, a window will open to download the attendance report as show below:

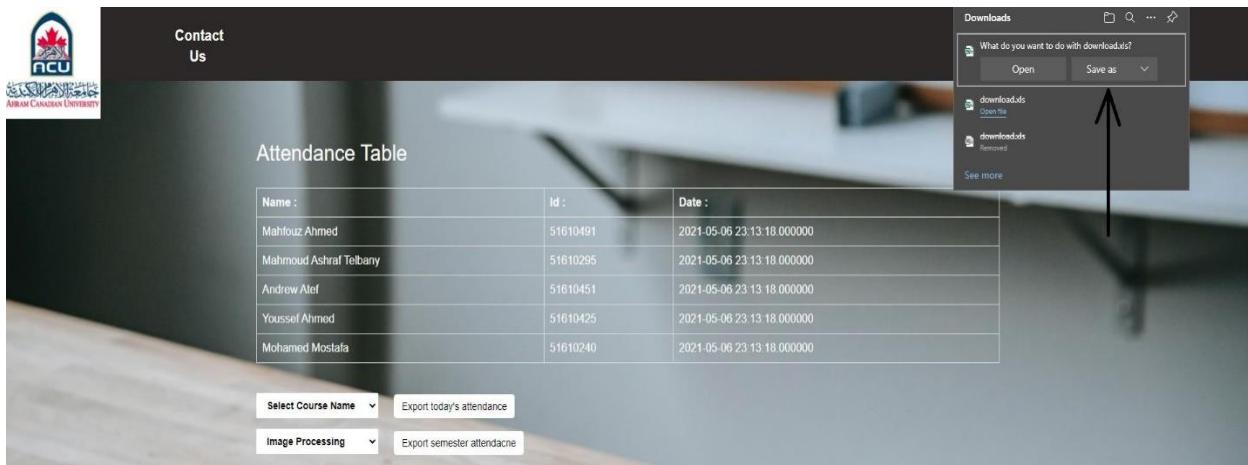


Fig. 3.20: Downloading the excel sheet

After downloading the excel file for the total attendance it will look like form in the figure 3.21.

The screenshot shows an Excel spreadsheet with the following data:

	A	B	C
1	name	id	count
2	Mahmoud Ashraf Telbany	51610295	3
3	Youssef Ahmed	51610425	1
4	Andrew Atef	51610451	4
5	Mahfouz Ahmed	51610491	2
6			

Fig. 3.21: Excel sheet contains the count of total attendance for each student

The PHP code for both downloading current lecture attendance and overall semester attendance are shown in figure 3.22, figure 3.23, and figure 3.24 respectively.

```
1  <?php
2  //export.php
3
4  $connect = mysqli_connect("localhost", "root", "", "qr");
5  $output = '';
6  if(isset($_POST["export"]))
7  {
8      $query = "SELECT * FROM student";
9      $result = mysqli_query($connect, $query);
10     if(mysqli_num_rows($result) > 0)
11     {
12         $output .= '
13             <table class="table" bordered="1">
14                 <tr>
15                     <th>name</th>
16                     <th>id</th>
17                     <th>date</th>
18
19
20                 </tr>
21             ';
22         while($row = mysqli_fetch_array($result))
23         [
24             $output .= '
25                 <tr>
26                     <td>' . $row["name"] . '</td>
27                     <td>' . $row["id"] . '</td>
28                     <td>' . $row["date"] . '</td>
29
30
31                 </tr>
32             ';
33         ]
34         $output .= '</table>';
35         header('Content-Type: application/csv');
36         header('Content-Disposition: attachment; filename=download.xls');
37         echo $output;
38     }
39 }
40 ?>
41
```

Fig. 3.22: PHP code for today's Attendance table download button

```

export1.php > ...
1  <?php
2
3
4  $connect = mysqli_connect("localhost", "root", "", "qr");
5  $output = '';
6  if(isset($_POST["export"]))
7  {
8      $fullName=$_POST['fullName'];
9      $query = "SELECT * FROM $fullName";
10     $result = mysqli_query($connect, $query);
11     if(mysqli_num_rows($result) > 0)
12     {
13         $output .= '
14             <table class="table" bordered="1">
15                 <tr>
16                     <th>name</th>
17                     <th>id</th>
18                     <th>count</th>
19
20
21                 </tr>
22         ';
23         while($row = mysqli_fetch_array($result))
24         {
25             $output .= '
26                 <tr>
27                     <td>'.$row["name"].'</td>
28                     <td>'.$row["id"].'</td>
29                     <td>'.COUNT($row["id"]).'</td>
30
31                 </tr>
32         ';
33     }
34
35     $output .= '</table>';
36     header('Content-Type: application/csv');
37     header('Content-Disposition: attachment; filename=download.xls');
38     echo $output;
39 }
40 }

```

Fig. 3.23: PHP code for semester Attendance export button

```

else{
    $reg=" INSERT INTO $fullName (name, id, ip, date) VALUES('$name','$id','$IP','$date') ";
    $regg="INSERT INTO `advanced_total_attendance` (SELECT * FROM advanced)";
    $reggg="INSERT INTO `image_total_attendance` (SELECT * FROM image)";
    $regggg="INSERT INTO `pattern_total_attendance` (SELECT * FROM pattern)";
    $reggggg="INSERT INTO `security_total_attendance` (SELECT * FROM security)";
    mysqli_query($conn,$reg);
    mysqli_query($conn,$regg);
    mysqli_query($conn,$reggg);
    mysqli_query($conn,$regggg);
    mysqli_query($conn,$reggggg);
    header('location:sss.html');
}

?>

```

Fig. 3.24: Continue of PHP code for semester Attendance export button

3.2 Mobile Application Part

The mobile application consists of different parts that union with each other to form a cooperation processing for different users such as:

- Login page for the instructor or admin login.
- Student login and scanner for QR code.
- Login history for specific user when logged in.
- Admin access for every course.

3.2.1 Authentication in Mobile Application

All of these parts are depend on firebase platform which are using Firestore database for control, view, and add courses or users. In the following the mobile application function will highlight with screen shots. Firstly, Authentication database in firebase is shown in figure 3.25,

The screenshot shows the Firebase console's Authentication section. On the left, there's a sidebar with project settings like Build, Authentication, Firestore Database, and Storage. The main area is titled 'Authentication' and has tabs for 'Users', 'Sign-in method', 'Templates', and 'Usage'. Below the tabs is a search bar and a 'Add user' button. A table lists six users with their email addresses, provider (Email), creation date, sign-in date, and user UID. The table includes columns for Identifier, Providers, Created, Signed In, and User UID. The user UIDs are partially obscured.

Identifier	Providers	Created	Signed In	User UID
51610295@acu.com	Email	May 16, 2021	Jun 27, 2021	MwEjonwaCYO25sGA5h9qKIMglg...
51610240@acu.com	Email	May 16, 2021	Jun 27, 2021	QrcX0mXh28b7InZu65QypClwKZ2
51610491@acu.com	Email	Jun 11, 2021	Jun 27, 2021	RwV4MhV7E9RBCh201b9fUddKy...
51610425@acu.com	Email	May 16, 2021	Jun 11, 2021	doBYi3JE5VWE77jCIYJER05hWW...
admin123456@acu.com	Email	Jun 11, 2021	Jun 27, 2021	jphh2NldzLBZJKjLO4ntKwzoCR2
51610451@acu.com	Email	May 16, 2021	Jun 27, 2021	ybiw7QyNPvW0mkZ4nFkEusd7K...

Fig. 3.25: Authentication database in Firebase

In authentication, we are going to adding students by ID@acu.com and adding admins by adminID@acu.com.

Secondly, cloud firestore is a flexible and scalable database for mobile & server development from Firebase in addition to Google cloud. Like Firebase Real-time Database, it keeps your data in sync across client apps through Real-time listeners. In addition, it offers offline support for mobile and web so, it can build responsive applications that work regardless of network latency or Internet connectivity. Cloud Firestore also offers seamless integration with other Firebase and Google cloud products including cloud functions. Its Key features capabilities are:

Flexibility means: The cloud Firestore data model supports flexible, hierarchical data structures. Store your data in documents, organized into collections. Documents can contain complex nested objects in addition to subcollections.

Expressive querying means: In cloud Firestore, you can use queries to retrieve individual, specific documents or to retrieve all the documents in a collection that match your

query parameters. Your queries can include multiple, chained filters and combine filtering and sorting. They're also indexed by default, so query performance is proportional to the size of your result set, not your data set.

Realtime updates means: like real-time database, cloud Firestore uses data synchronization to update data on any connected device. However, it's also designed to make simple, one-time fetch queries efficiently.

Offline support means: cloud Firestore caches data that your app is actively using, so the app can write, read, listen to, and query data even if the device is offline. When the device comes back online, cloud Firestore synchronizes any local changes back to cloud Firestore.

Designed to scale means: cloud Firestore brings you the best of Google cloud's powerful infrastructure: automatic multi-region data replication, strong consistency guarantees, atomic batch operations, and real transaction support. It has designed cloud Firestore to handle the toughest database workloads from the world's biggest apps.

How does cloud Firestore work?

Cloud Firestore is a cloud-hosted, NoSQL database that your iOS, Android, and web apps can access directly via native SDKs. Cloud Firestore is also available in native Node.js, Java, Python, Unity, C++ and Go SDKs, in addition to REST and RPC APIs.

Following cloud Firestore's NoSQL data model, you store data in documents that contain fields mapping to values. These documents are stored in collections, which are containers for your documents that you can use to organize your data and build queries. Documents support many different data types, from simple strings and numbers, to complex, nested objects. You can also create sub-collections within documents and build hierarchical data structures that scale as your database grows. The cloud Firestore data model supports whatever data structure works best for your app.

Additionally, querying in cloud Firestore is expressive, efficient, and flexible. Create shallow queries to retrieve data at the document level without needing to retrieve the entire collection, or any nested sub-collections. Add sorting, filtering, and limits to your queries or cursors to paginate your results. To keep data in your apps current, without retrieving your entire

database each time an update happens, add real-time listeners. Adding real-time listeners to your app notifies you with a data snapshot whenever the data your client apps are listening to changes, retrieving only the new changes.

Protect access to your data in cloud Firestore with Firebase authentication and cloud Firestore security rules for Android, iOS, and JavaScript, or Identity and Access Management (IAM) for server-side languages. Figure 3.26 show cloud Firestore database structure.

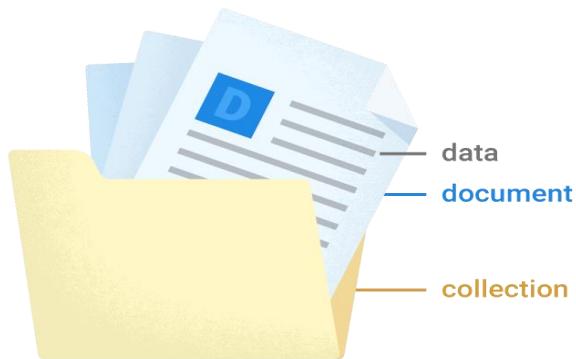


Fig. 3.26: Cloud Firestore database structure

3.2.2 Student History in Mobile Application

Now we are using Firestore database to add document for every student ID and set his/her name, ID data and also add courses counter to count times of scanning QR code by student for specific course as shown in figure 3.27.

Fig. 3.27: Document for every student ID

Also, a location function has been added. This function helps in student locating when logged in by our application by taking his/her geohash and geopoint as shown in figure 3.28.

Fig. 3.28: Show location

Every student has student history which recording every QR- code scanning times performed by this student as shown in figure 3.29.

The screenshot shows the Firebase Cloud Firestore interface. On the left, the navigation sidebar includes sections for Build (Authentication, Firestore Database, Realtime Database, Storage, Hosting, Functions, Machine Learning), Release & Monitor (Crashlytics, Performance, Test Lab), Analytics (Dashboard, Realtime, Events, Conv.), and Finance (Extensions). The main area displays a document in the 'History' collection under 'Users'. The document ID is ZfRbHk18BfaHnCMUEP4F. It contains fields such as 'Courses Counter' (Advanced Databases: 2, Security: 1), 'Last Joined' (Course: "Advanced Databases"), 'Last Login' (June 27, 2021 at 7:15:51 PM UTC+2), 'Last Scan' (June 11, 2021 at 5:10:29 PM UTC+2), and a QR code image.

Fig. 3.29: Student QR-code scanning history

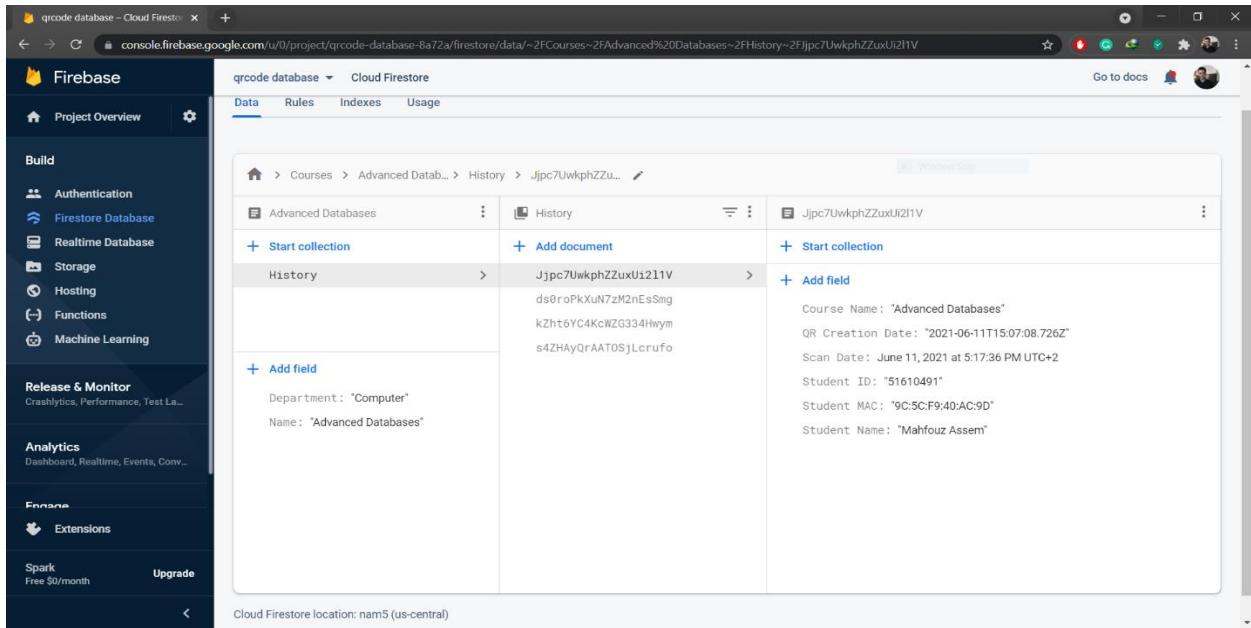
3.2.3 Courses History in Mobile Application

We are adding new document for every course data. Also, implement courses names and which department this course is belonging to as shown in figure 3.30.

The screenshot shows the Firebase Cloud Firestore interface. The navigation sidebar is identical to Fig. 3.29. The main area displays a document in the 'Courses' collection under 'qr-code-database-8a72a'. The document ID is Advanced Databases. It contains fields 'Image Processing', 'Math4', 'Pattern Recognition', 'Security', 'Department' ("Computer"), and 'Name' ("Advanced Databases").

Fig. 3.30: Courses document

Every course has its own history collection by recording every QR-code scanning by every student registered to that course as shown in figure 3.31.



The screenshot shows the Firebase Cloud Firestore interface. On the left, a sidebar lists various services: Authentication, Firestore Database, Realtime Database, Storage, Hosting, Functions, and Machine Learning. Below that are sections for Release & Monitor and Analytics. The main area is titled 'Cloud Firestore' and shows a hierarchical path: Home > Courses > Advanced Databases > History > Jjpc7UwkphZZuxUi211V. The 'Data' tab is selected. In the center, there's a table with three columns: 'Advanced Databases' (with '+ Start collection'), 'History' (with '+ Add document'), and 'Jjpc7UwkphZZuxUi211V' (with '+ Start collection'). Under 'History', there's a single document with fields: 'Department: "Computer"' and 'Name: "Advanced Databases"'. To the right, detailed information for the document 'Jjpc7UwkphZZuxUi211V' is displayed, including: Course Name: "Advanced Databases", QR Creation Date: "2021-06-11T15:07:08.726Z", Scan Date: "June 11, 2021 at 5:17:36 PM UTC+2", Student ID: "51610491", Student MAC: "9C:5C:F9:40:AC:9D", and Student Name: "Mahfouz Assem".

Fig. 3.31: History of courses

3.2.4 Controlling Pages in Mobile Application

Finally, login page of attendance system in the part of mobile application is shown in figure 3.32. The main page of the mobile application after login by user ID and password is shown in figure 3.33.

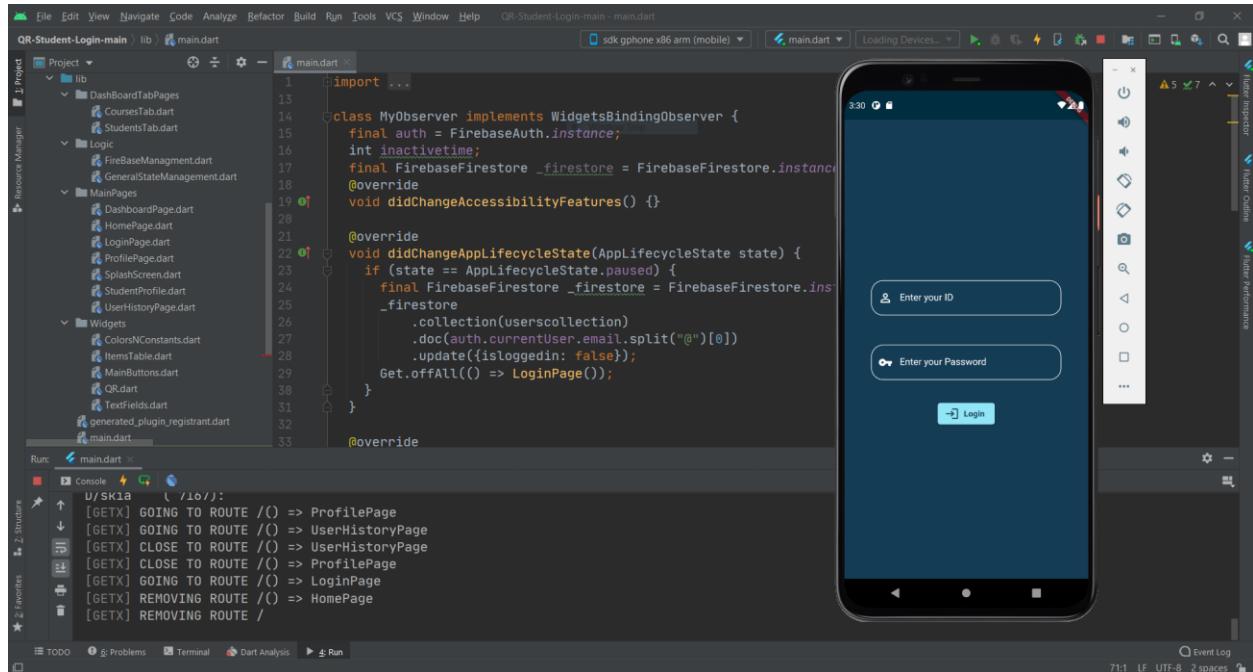


Fig. 3.32: Mobile application login page

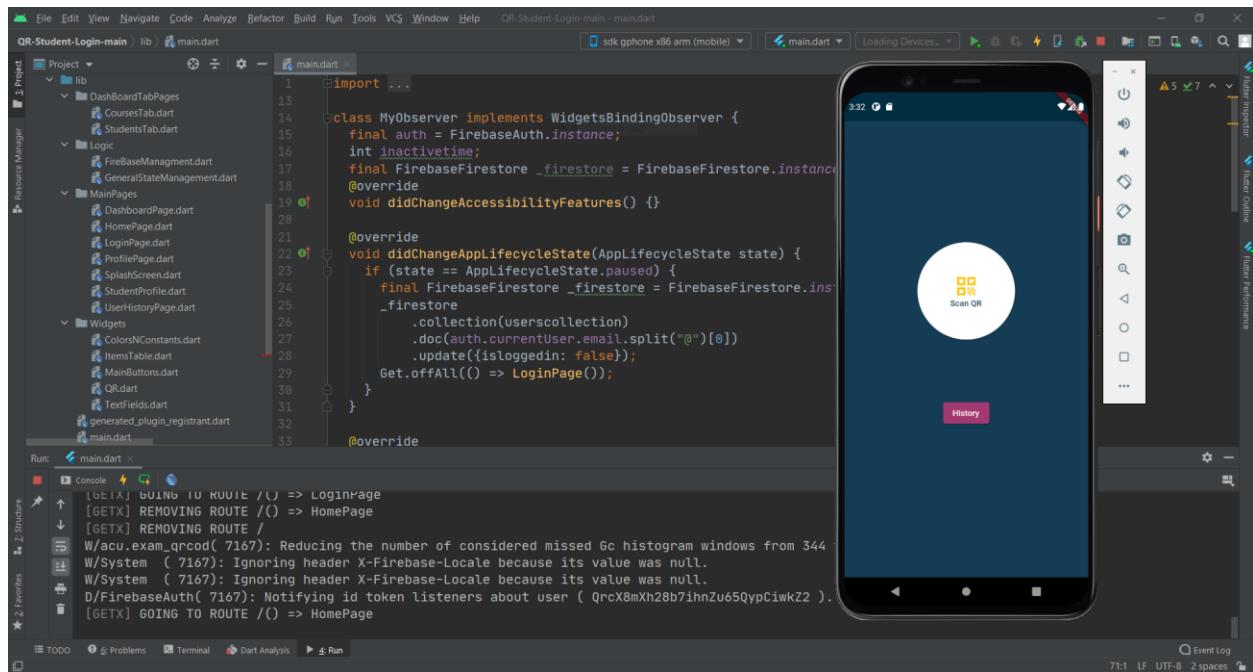


Fig. 3.33: Main page of mobile application after login

Every user has its profile info view which contains some details like user name, user ID, last scan, login date, last joined course, and courses counter. The user profile page is shown in figure 3.34.

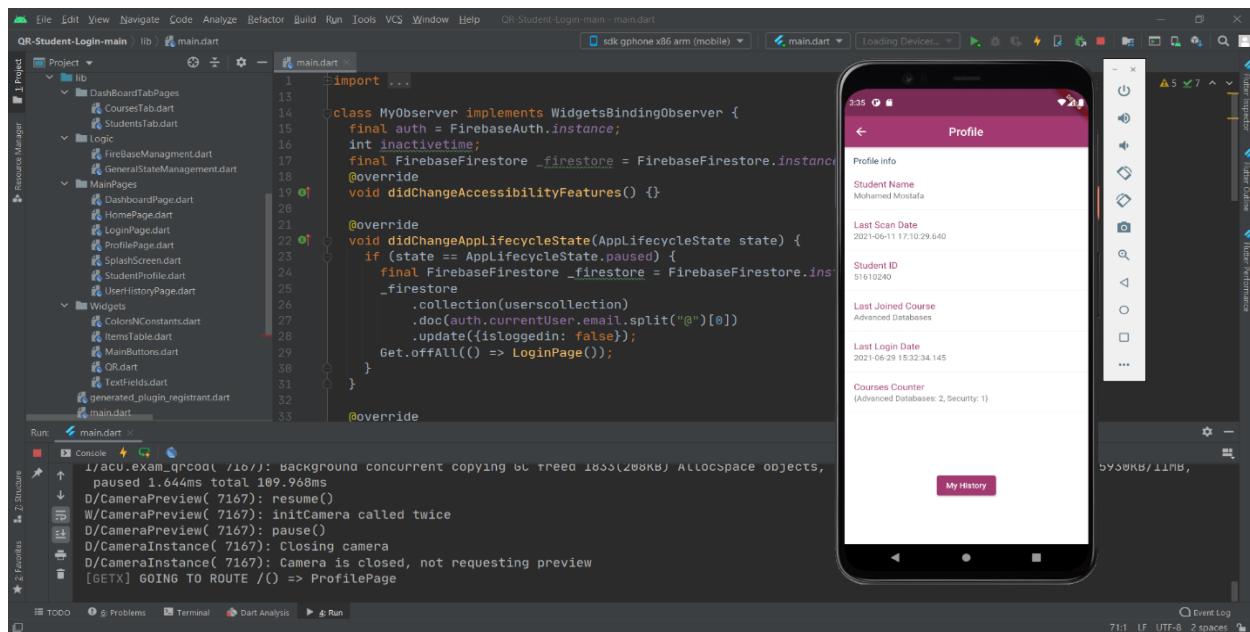


Fig. 3.34: Mobile application user profile page

When a student selects to show his history, the application displays some details like QR creation date, scan date, login date and course name just like shown in figure 3.35 and figure 3.36.

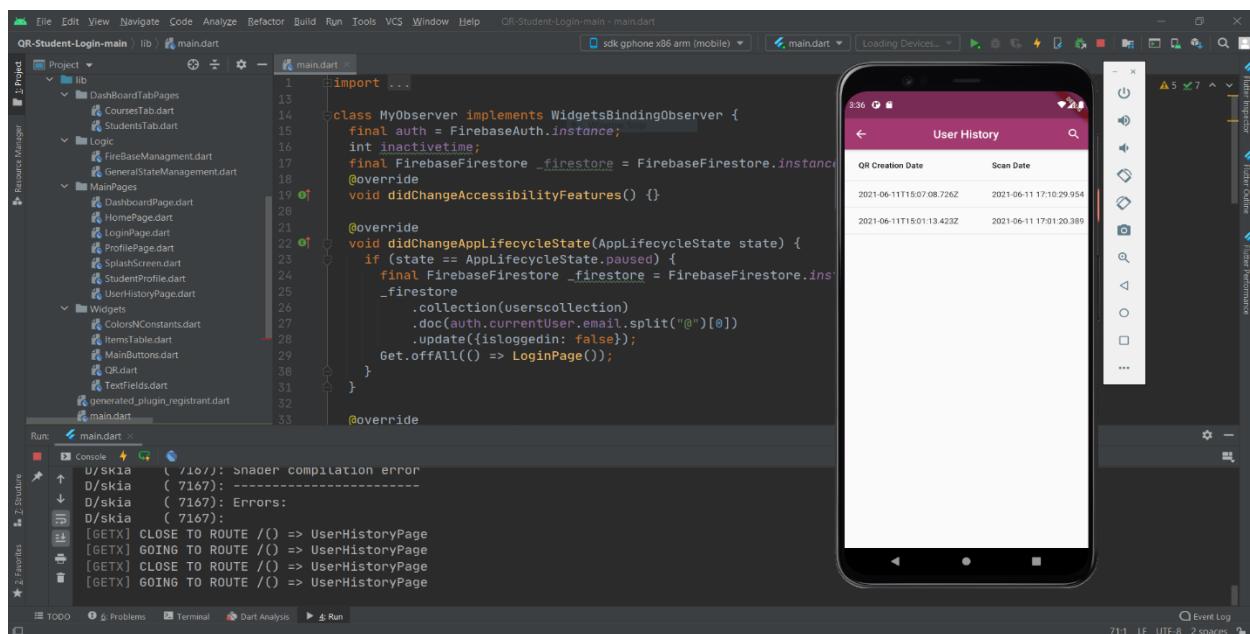


Fig. 3.35: User history page one

```

import ...
class MyObserver implements WidgetsBindingObserver {
  final auth = FirebaseAuth.instance;
  int inactivetime;
  final FirebaseFirestore _firebase = FirebaseFirestore.instance;
  @override
  void didChangeAccessibilityFeatures() {}

  @override
  void didChangeAppLifecycleState(AppLifecycleState state) {
    if (state == AppLifecycleState.paused) {
      final FirebaseFirestore _firebase = FirebaseFirestore.instance;
      _firebase
        .collection('userscollection')
        .doc(auth.currentUser.email.split('@')[0])
        .update({'isloggedin': false});
      Get.offAll(() => LoginPage());
    }
  }

  @override
}

```

Fig. 36: User history page two

When admin logged in, he can access all data in firestore database such as displays all students, all courses and can search by student ID and course full name as shown in figure 3.37 which show the admin dashboard.

```

import ...
class MyObserver implements WidgetsBindingObserver {
  final auth = FirebaseAuth.instance;
  int inactivetime;
  final FirebaseFirestore _firebase = FirebaseFirestore.instance;
  @override
  void didChangeAccessibilityFeatures() {}

  @override
  void didChangeAppLifecycleState(AppLifecycleState state) {
    if (state == AppLifecycleState.paused) {
      final FirebaseFirestore _firebase = FirebaseFirestore.instance;
      _firebase
        .collection('userscollection')
        .doc(auth.currentUser.email.split('@')[0])
        .update({'isloggedin': false});
      Get.offAll(() => LoginPage());
    }
  }

  @override
}

```

Fig. 37: Admin dashboard page

Admin can access student profile by click on his/her specific name as shown in the following:

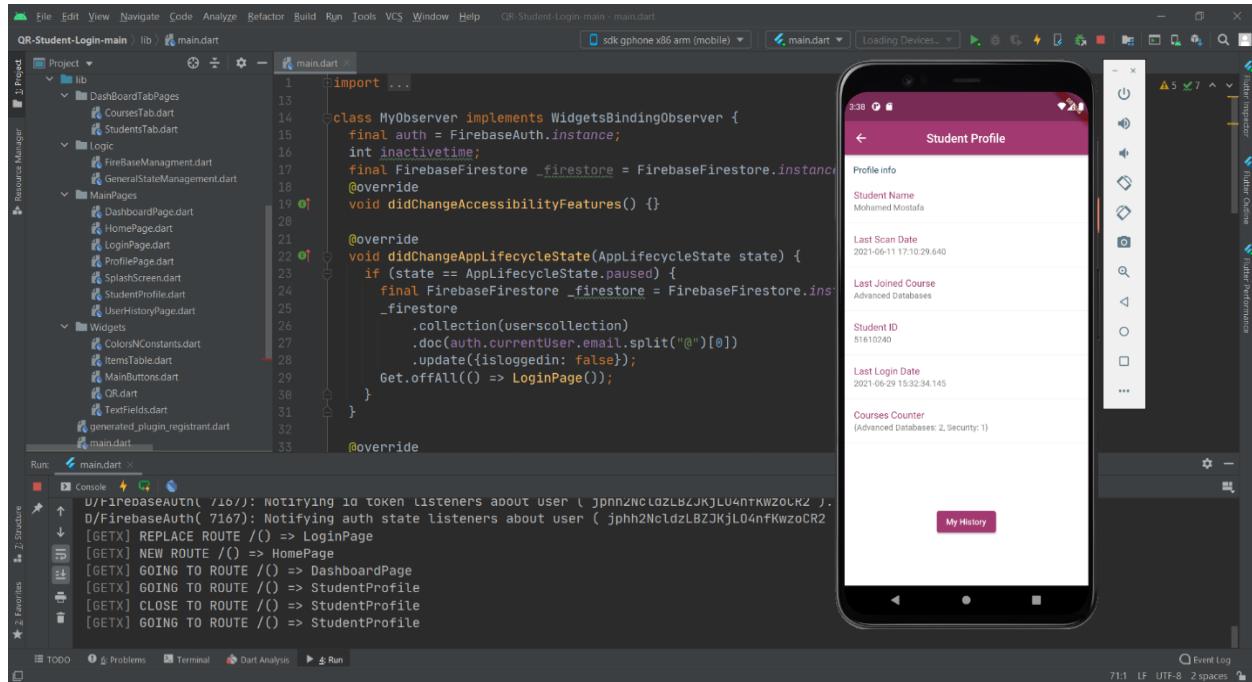


Fig. 3.38: Access of student data by admin login

Now admin can access courses and see all courses details as shown in the following

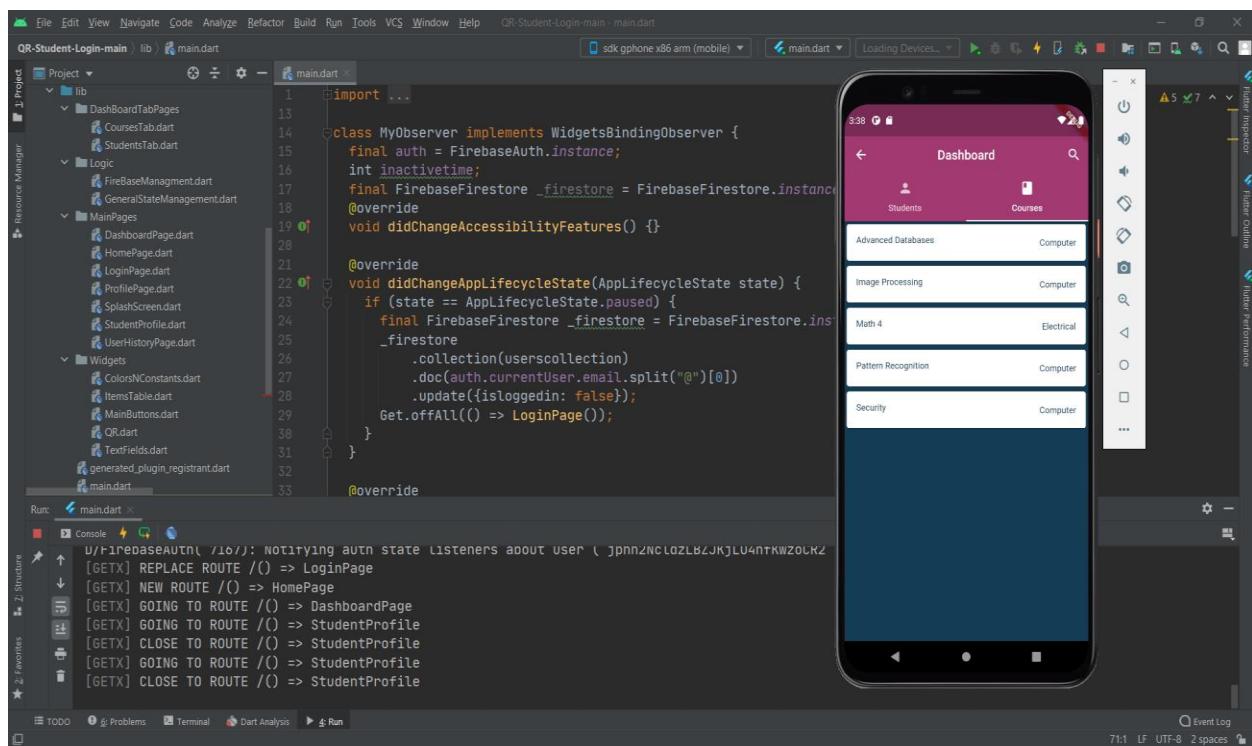


Fig. 3.39: Courses information page

When admin select a specific course, the history of students who scanned QR code of this course will be displayed as shown in figure 3.40.

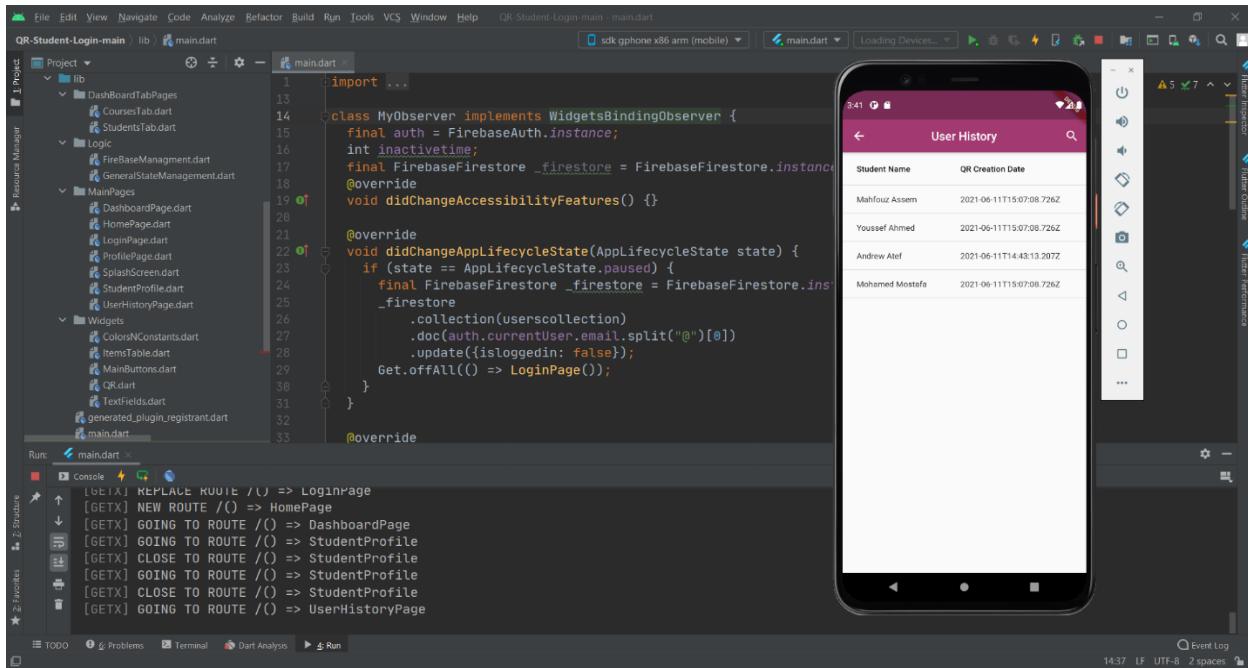


Fig. 3.40: Course history contains students scanned QR code

Also, admin can see Student ID, Scan date, and Student ID as shown in the following:

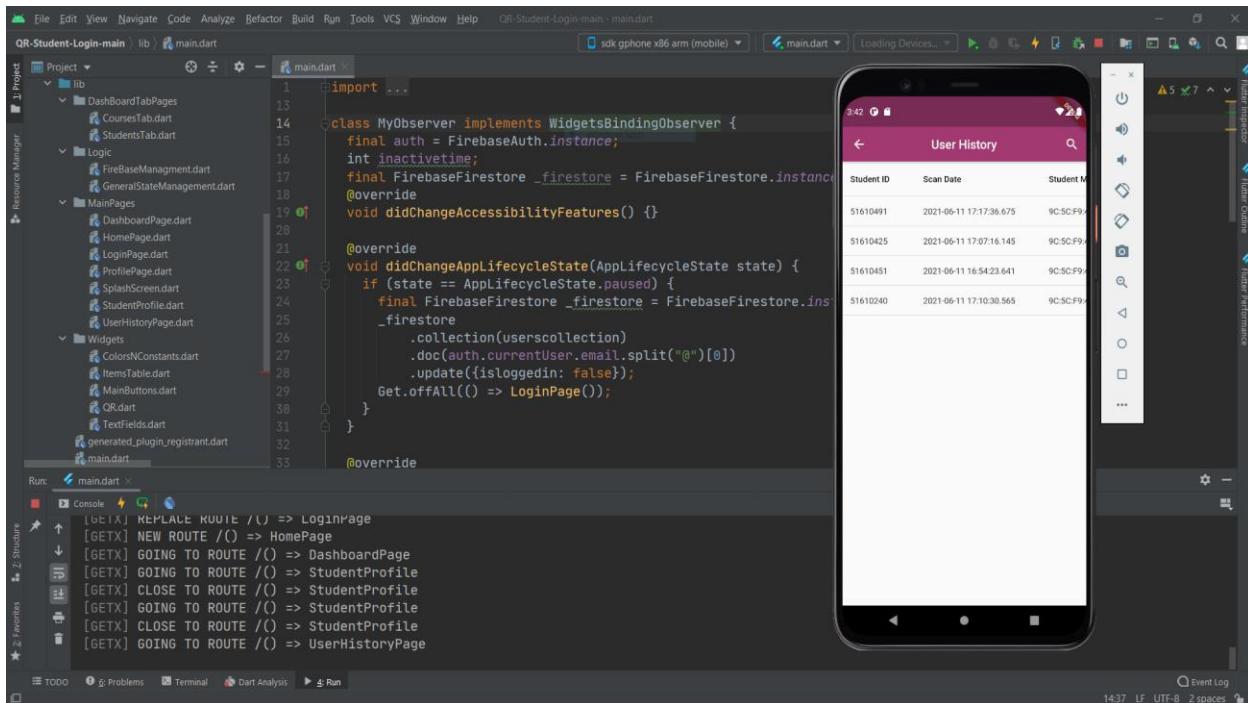


Fig. 3.41: The history of student's information scanned QR code

Also, admin can display MAC address of all devices which each student used to scan a specific course name as shown in the following:

The screenshot shows the Android Studio interface with the following components:

- Project View:** Shows the project structure under the lib folder, including DashBoardTabPages, CoursesTab.dart, StudentTab.dart, Logic, MainPages, and Widgets.
- Code Editor:** Displays the main.dart file with Dart code. The code includes imports, a MyObserver class that implements WidgetsBindingObserver, and overridden methods for didChangeAccessibilityFeatures and didChangeAppLifecycleState. It also contains logic related to FirebaseAuth and Firestore.
- Run Tab:** Shows the current routes: [GET] REPLACE ROUTE /() => LoginPage, [GET] NEW ROUTE /() => HomePage, [GET] GOING TO ROUTE /() => DashboardPage, [GET] GOING TO ROUTE /() => StudentProfile, [GET] CLOSE TO ROUTE /() => StudentProfile, [GET] GOING TO ROUTE /() => StudentProfile, [GET] CLOSE TO ROUTE /() => StudentProfile, and [GET] GOING TO ROUTE /() => UserHistoryPage.
- Preview:** A smartphone preview shows the "User History" screen with a table listing student MAC addresses and course names. The table data is as follows:

Student MAC	Course Name
9C:5C:F9:40:AC:90	Advanced Databases

- Toolbars and Panels:** Includes the Device Manager, Run, Build, and VCS toolbars, along with various inspection and performance analysis panels on the right side.

Fig. 3.42: The history of users scanned QR code info

PROJECT PLANNING REPORT

Project Name: QR Attendance System							
Prepared By: (Alphabetical order)	Date:	Reporting Period: 18/11/2020 to 3/7/2021					
Ahmed El Sayed Andrew Atef Mahfouz Assem Mahmoud El Telbany Mohamed Mostafa Mostafa Mahdy Yara Hossam Youssef Ahmed	3/7/2021						
Project Overall Status: Done.							
<p>Project Summary:</p> <p>The QR Attendance System decreases the time taken in taking attendance with the traditional way as every student has a smartphone will download this application, can take and know how many days they attend as well as the teacher can log in to calculate the number of attendance day for each student. The QR code is very secured as it is difficult for a student to attend another student.</p>							
Milestone Deliverables performance reporting over last period							
Milestone Deliverables	Due Date	% Completed	Deliverable Status	Done By			
Proposal Design							
Description of the System's environment Overview.	18/11/2020	100%	[On Schedule]	Ahmed El Sayed Andrew Atef Mahfouz Assem Mahmoud El Telbany Mohamed Mostafa Mostafa Mahdy Yara Hossam Youssef Ahmed			
Conceptual Design							
Description of the System's environment: General description of the environment including all its players, related processes, problems, limitations, etc	18/11/2020	100%	[On Schedule]	Ahmed El Sayed Andrew Atef Mahfouz Assem Mahmoud El Telbany Mohamed Mostafa Mostafa Mahdy Yara Hossam Youssef Ahmed			

<ul style="list-style-type: none"> Identification of the system's needs: What are the general needs? Who is the customer (or groups of customers)? What are the goals of the customers? What are the customers' systems' requirements? What are the possible technical design elements? 	13/1/2021	100%	[On Schedule]	Ahmed El Sayed Andrew Atef Mahfouz Assem Mahmoud El Telbany Mohamed Mostafa Mostafa Mahdy Yara Hossam Youssef Ahmed
Is there a need/possibility to use tools like QFD to develop those needs in details? If so, can you give an example?!	13/1/2021	100%	[On Schedule]	Ahmed El Sayed Andrew Atef Mahfouz Assem Mahmoud El Telbany Mohamed Mostafa Mostafa Mahdy Yara Hossam Youssef Ahmed
Operations and performance requirements: What are they? What, when, where, how? Identify the main functions of the systems and their relationships. What are the technical performance measures (TPMs)? Can you identify design dependent parameters (DDPs)? Is there a legacy system that needs to be considered? Can you generate System's Specifications at this point? !	13/1/2021	100%	[On Schedule]	Ahmed El Sayed Andrew Atef Mahfouz Assem Mahmoud El Telbany Mohamed Mostafa Mostafa Mahdy Yara Hossam Youssef Ahmed
Maintenance and support Requirements: Can you identify them at this stage?!	21/1/2021	100%	[On Schedule]	Ahmed El Sayed Andrew Atef Mahfouz Assem Mahmoud El Telbany Mohamed Mostafa Mostafa Mahdy Yara Hossam Youssef Ahmed

<p>Conceptual design decisions: From all previous sections, can you identify various options that could be used for the systems? These include human, machines and software options. Based on experience make a discussions of tradeoffs and possibilities</p>	21/1/2021	100%	[On Schedule]	<p>Ahmed El Sayed Andrew Atef Mahfouz Assem Mahmoud El Telbany Mohamed Mostafa Mostafa Mahdy Yara Hossam Youssef Ahmed</p>
Preliminary Design				
<p>Translate now all conceptual design stage data into detailed Systems' functions and their relationships. For as many functions as possible detail the function including its inputs, controls and constraints, internal mechanisms and output. Be organized in the functional hierarchy thinking from high level functions to their lower level elements.</p>	28/1/2020	100%	[On Schedule]	<p>Ahmed El Sayed Andrew Atef Mahfouz Assem Mahmoud El Telbany Mohamed Mostafa Mostafa Mahdy Yara Hossam Youssef Ahmed</p>
<p>What are the various professional disciplines needed to perform the detailed design and/or development of the functions? How will you involve those disciplines? At what cost?!</p>	28/1/2021	100%	[On Schedule]	<p>Ahmed El Sayed Andrew Atef Mahfouz Assem Mahmoud El Telbany Mohamed Mostafa Mostafa Mahdy Yara Hossam Youssef Ahmed</p>
<p>Think well of the resources needed for each function. There may be various options to choose from. Make a tradeoff analysis if possible.</p>	28/1/2021	100%	[On Schedule]	<p>Ahmed El Sayed Andrew Atef Mahfouz Assem Mahmoud El Telbany Mohamed Mostafa Mostafa Mahdy Yara Hossam</p>

				Youssef Ahmed
Detailed Design				
Choose one or more functions that you can design in details. This could be a detailed design if it's in your field of expertise or you could just detail all the disciplines needed for its detailed design.	31/1/2021	100%	[On Schedule]	Ahmed El Sayed Andrew Atef Mahfouz Assem Mahmoud El Telbany Mohamed Mostafa Mostafa Mahdy Yara Hossam Youssef Ahmed
Project Documentation				
Writing the project documentation	13/1/2021	100%	[On Schedule]	Ahmed El Sayed Andrew Atef Mahfouz Assem Mahmoud El Telbany Mohamed Mostafa Mostafa Mahdy Yara Hossam Youssef Ahmed
Project Presentation				
Preparing the project presentation	3/7/2021	100%	[On Schedule]	Ahmed El Sayed Andrew Atef Mahfouz Assem Mahmoud El Telbany Mohamed Mostafa Mostafa Mahdy Yara Hossam Youssef Ahmed

REFERENCES:

- [Al Hajry, 2019] Al Hajri, Eid; Hafeez, Farrukh; and Azhar, Ameer (2019). Fully automated classroom attendance System. In International Journal of Interactive Mobile Technologies (iJIM), Vol. 13, No. 8.
- [Almasalha, 2014] Almasalha, Fadi & Hirzallah, Nael. (2014). A Students Attendance System Using QR Code. International Journal of Advanced Computer Science and Applications.
- [Ayoub, 2018] Ayop, Zakiah & Yee, Chan & Anawar, Syarulnaziah & Erman, Hamid & Syahrul, Muhammad. (2018). Location-aware Event Attendance System using QR Code and GPS Technology. International Journal of Advanced Computer Science and Applications.
- [Baban, 2014] Baban, Miran Hikmat. (2014) Attendance cheching system using Quick Response code for students at the university of Sulaimaniyah.Journal of Mathematics and Computer Science. Vol. 10. 189-198
- [Change, 2014] Chang, Jae. (2014). An introduction to using QR codes in scholar journals. Science Editing. Vol.1, 113-117.
- [Kumar, 2017] Kumar, B. Dinesh and Kareemula, S. (2017) Smart mobile attendance system for employees using QR scanner. Assian Journal of Applied Science and Technology (AJAST) Vol.1, No 5, 35-39.
- [Gupta, 2020] Gupta, [Naman](#); Sharma, [Purushottam](#); Deep, [Vikas](#); and Shukla,[Vinod Kumar](#). (2020). Automated attendance system using OpenCV. In proceedings of 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO).
- [Singh, 2019] Singh, Gurlove; Dwivedi, Rohit; A nand, A bhineet. (2019). Attendance monitoring and management using QR code based sensing with cloud based processing. International Journal of Scientific Research in Computer Science Applications and Management Studies, 2019