



Smart Classroom using Raspberry Pi

project graduation report submitted in partial fulfilment for the award of BSc.
Degree in Electrical Engineering, Benha Faculty of Engineering, Benha University

Presented by:

1. Mohamed Mohamed Abdelhafez Tantawy.
2. Hossam Ashraf Abdelfatah Ahmed youseff.
3. Abdelrahman Saber Abdelfatah Mostafa.
4. Adelrahman Mohamed Gad elazab.
5. Mohamed Elsayed Abdelsalam Mobarak.
6. Hesham Abdalah nourelden Mohamed Ahmed.
7. Mahmoud Mohamed Gouda metwaly.

Supervised by:

Dr: Sara Hamdy.

Eng: Ahmed Ragab.

Benha - Egypt

July 2022

Table of contents

Contents

A b s t r a c t:	9
1. Chapter One: Introduction	10
1.1 Learning	11
1.2 Attendance methodology	17
1.3 Rapid growth of smart devices	18
1.4 Conclusions.....	21
2. Chapter Two: IOT.....	22
2.1 Introduction.....	22
2.2 Internet of Everything (IoE):	23
2.3 IoT value in different settings.....	24
2.4 Project interest	35
2.5 Conclusion	36
3. Chapter Three: Raspberry pi.....	37
3.1 Introduction.....	37
3.2 Raspberry Pi hardware specifications.....	38
3.3 Raspberry Pi 4 B	55
3.3.1 LCD:	57
3.3.2 Install the touch driver.....	59
3.3.3 Screen orientation settings.....	61
3.3.4 Touch screen calibration.....	62
3.3.5 Install Virtual Keyboard	63
3.3.6 Raspberry pi Camera:	68
3.4 Implementation:	71
3.5 Main_Module:	71
3.5.1 Websocket protocol.....	71

3.6 WebSocket implementation in our project:	75
.....	76
3.7 Exam:	83
3.8 Facial recognition:	97
3.9 Train model:.....	102
3.10 Video_pictures:.....	104
3.11 Voting:	108
3.12 Conclusion	112
4. Chapter Four: Web Application front end.....	113
4.1 Introduction.....	113
4.2 why using html?.....	113
4.3 The webpages made in our project	115
4.4- The exam part	187
4.5- How all pages are connected to the server?	187
4.6 Conclusion	187
5. Chapter Five: WebSocket.....	188
5.1 Introduction:.....	188
5.2 WebSocket:.....	189
5.3 Long polling	190
5.4 SocketIO	194
5.5 XMPP.....	197
5.4.1 How Watsapp use XMPP	198
5.5 WebSocket Implementation.....	203
5.6 Conclusion and future considerations.....	207
6. Chapter six: Backend application	209
6.1 Introduction:	209
6.2 Access and modify this data	210
6.3 Relational database	211

6.4 Non-relational database.....	213
6.5 Database schema.....	214
6.6 DBMS:	222
6.7 Back-End Web Architecture	225
6.8 Server vs. Serverless: Which Is the Best Option?	229
6.9 Web Application Framework	235
6.10 Conclusion	239
7. Chapter Seven: Mobile signal detector.....	240
7.1 Mobile signal	240
7.2 Detector is valuable	244
7.3 Signal detector	246
7.4 (Feedback/Capacitor) Detector	251
Conclusion and Future work.....	267
References	269
الملخص	273

List of figures

Figure 1-1. Smart Home	11
Figure 1-2.Comparison of different types of learning	20
Figure 1-3. Advantage of Smart learning via IOT	20
Figure 2-1. value potential of the Internet of Things.....	25
Figure 2-2. iot-healthcare-wearable-heart-race-tracker	27
Figure 2-3.real-time-for-early-detection-of-heart-diseases	28
Figure 2-4.Various IoT applications in homes	29
Figure 2-5.Examples of IoT uses in retail.....	32
Figure 3-1. Raspberry Pi 2 module	38
Figure 3-2. Raspberry Pi models	54
Figure 3-3. Raspberry Pi 4 Front view	57
Figure 3-4. LCD model.....	58
Figure 3-5. LCD	58
Figure 3-6: HTTP communication.....	73
Figure 5-1. HTTP Protocol	189
Figure 5-2. Long Polling.....	192
Figure 5-3. Network layers	193
Figure 5-4. WebSoket and Socket	194
Figure 5-5.WebSocket	196
Figure 5-6. comparison between WebSocket and Socket.IO	196
Figure 5-7. WhatsAPP Registration Flow.	201
Figure 6-1. Database schema	215
Figure 6-2: DBMS Architecture	217
Figure 6-3: Data model	218
Figure 6-4. Data	218
Figure 6-5.Mainframe environment.....	219
Figure 6-6. Client/Server environment	220
Figure 6-7. Client/Server environment	221
Figure 6-8. physical model for our project	222
Figure 6-9. Popular Backend frameworks	234
Figure 6-10. frontend & backend frame works.....	237
Figure 7-1. waves	241
Figure 7-2. Ionosphere reflection.....	243
Figure 7-3. Traditional communication	244
Figure 7-4. LED	246

Figure 7-5. Types of antenna	248
Figure 7-6. Simple cell phone detector circuit.....	250
Figure 7-7. Simple cell phone detector circuit 2.....	251
Figure 7-8. CA313 IC (OP-AMP)	253
Figure 7-9. Data sheet of CA313 IC (OP-AMP)	253
Figure 7-10. Transistor BC-548 Datasheet	257
Figure 7-11. Two 2.2 M Ω Resistors	257
Figure 7-12. 1K Resistor.....	258
Figure 7-13. 100 K Ω Resistor.....	258
Figure 7-14. 47 pF Capacitor	258
Figure 7-15. 0.22 uF Capacitor	258
Figure 7-16. 100 uF (Chemical).....	259
Figure 7-17. 0.47 uF (Chemical).....	259
Figure 7-18. Circuit Diagram of mobile detector.	262
Figure 7-19. Circuit on test Breadboard	262
Figure 7-21. Schematic diagram	263
Figure 7-20. Pcb circuit.....	263
Figure 7-22. Circuit on PCB	265

List of Tables

Table 2-1. IoT vs IoE	24
Table 2-2. IOT Software development Platforms.....	36
Table 3-1. Comparison between PC and Rasberry Pi.....	41
Table 3-2. Raspberry Pi models Comparison	55
Table 3-3. Raspberry Pi Pins Description.....	67
Table 5-1. Comparison between WebSocket and EventSource	190
Table 6-1. Server vs Serverless.....	232
Table 7-1. Comparison between inverting Op-amp & Non-inverting Op-amp	254
Table 7-2. Component and its Cost	266

List of Abbreviations

M2M	is machine to machine communication
P2P	people-to-people interactions
IoT	Internet of Things
CRSs	Classroom Response Systems
ASR	audience response system
DFD	Data Flow Diagram
FYP	Final year project
LDR	Light Dependent Resistor
IoE	Internet of Everything
TCP	Transmission Control Protocol
IIoT	Industrial Internet of Things
LAN	Local Area Network
PAN	Personal Area Network
RFID	Radio Frequency Identification
WAN	Wide Area Network
WS	WebSocket
UDP	User datagram protocol
DoS	Denial-of-Service
RF	radio frequency

A b s t r a c t:

With the advent of smart homes, smart cities, and smart everything, the Internet of Things (IoT) has emerged as an area of incredible impact, potential, and growth, with Cisco Inc. predicting to have 50billion connected devices by 2020.

Internet, a revolutionary invention, is always transforming into some new kind of hardware and software making it unavoidable for anyone. The form of communication that we see now is either human-human or human-device, but the Internet of Things (IoT) promises a great future for the internet where the type of communication is machine-machine (M2M).

We present Smart classroom is representative of modern teaching. With the advent of modern technology it becomes easier for the students as well as teachers to perform their task more efficiently. The smart classroom is rectification to various problems that teachers as well as students face inside the classroom. This project aims to make use of modern technology for helping the teachers in utilizing more time for teaching and students to easily get access to the study material. In this project we are building a prototype of a smart classroom by adding some tools that help the students and the lecturer and reduce time wasting during classes like face detection and recognition and detect him in exams and to get attendance and voting and send this data to a local server and detect the try of cheating if any student tried to make a call during exam and other tools, this application would be pivotal for carrying out various operations in the classroom.

1. Chapter One: Introduction

Teaching is an ancient phenomenon; there has always been certain practices and styles of teaching. With the new technological advancements, the teaching styles have evolved. However, the technology may also have adverse effects on learning. Therefore, there is a need to study impact of each new technology and corresponding style on overall learning. A technology can only be accepted if it is affecting the learning process positively. Consequently, there has been a large number of studies on the impact of various technologies on the learning outcome.

A *smart classroom* is defined as technology-assisted closed environment that enhances teaching and learning experience. A typical *smart classroom* has tools for better presentation, student engagement enhancement, better interaction, and better physical environment. It can also have tools for various logistic tasks such as taking attendance, assessment, and real-time feedback. In the literature, we identified four major facets of teaching and learning experience:

- (1) **Smart content:** This includes preparation, delivery, and distribution of the rich and interactive multimedia content.
- (2) **Smart interaction & engagement:** This includes interaction among students, the interaction between a teacher and a student, and engagement of student during the lecture time.
- (3) **Smart assessment:** This includes an assessment of students learning (quiz etc.) and feedback to the teacher (lecture quality).
- (4) **Smart physical environment:** Along with smart teaching and learning aids, a *smart classroom* should also have a healthy physical environment (temperature, humidity, etc.).

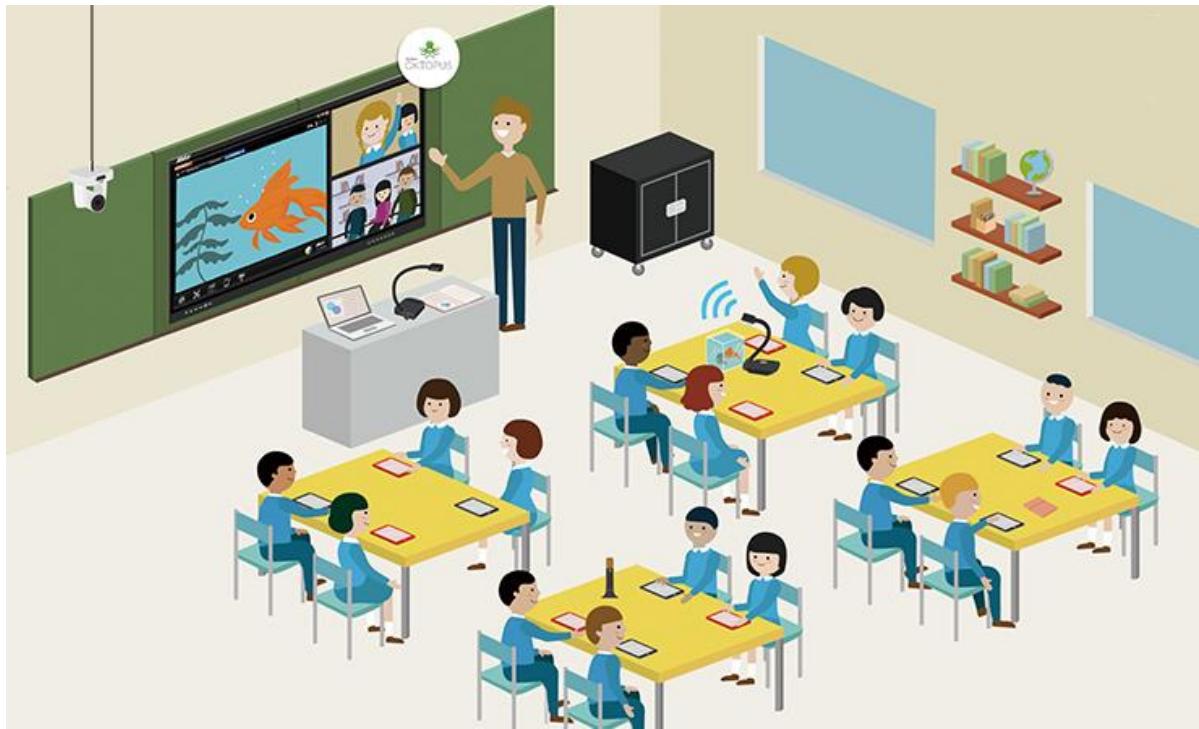


Figure 1-1. Smart Home

1.1 Learning

Learning analytics is defined as an analysis of student actions to assess academic progress and predict future performance. Student actions include online social interactions, exams, assignments, posts on discussion forums, etc.

Taking attendance is a time-consuming task. There has been a number of works on automatic attendance taking. RFID and Face recognition based student attendance solutions. Likewise, distance learning requires a number of synchronized tools for smooth conduct of lectures at local and remote classrooms.

Classroom Response Systems (CRSs) In a typical CRS, the students send their responses to the instructor electronically. The responses are gathered and analyzed by the instructor. A similar survey is conducted by Kay et al. [28] to assess the impact of audience response system (ASR). ASR is a type of CRS in which the questions are displayed electronically and the audience answer the questions using their mobile device. Witnessing the pervasive use of mobile computing devices - such as notebooks, tablets, and smartphones.

Smart classrooms have been a product of development and adoption of technology in the education sector. Traditionally, learning practically occurred in physical

classrooms where teachers and students came together to allow the learning process to take place.

In the event of a need for some practical experience, there was some assembly of some equipment in a separate room such as a laboratory and the teachers together with the students engaged in the practical experiments to supplement the skills and knowledge acquired in the classroom. Learning was only possible through physical contact of the students and the teachers.

Smart classrooms are the classrooms enhanced with technological equipment for the purpose of better learning and teaching. Based on the equipment available smart classrooms can be categorized as follows:

- **Basic Smart Classes:** The classrooms with basic smart technology include gadgets like laptops or computers, projector, DVD or VCD player and a viewing screen etc.
- **Intermediate Smart Classes:** The intermediate smart classrooms are one step ahead of the basic technology smart classrooms. They include gadgets like a smart podium with the control panel in addition to a laptop, projector, screen and DVD or VCD player etc.
- **Advanced Smart Classes:** The advanced smart classrooms have all the gadgets of a basic or intermediate smart classroom but their features are very advanced, that is, they use the newest technology.

What are the Equipments Installed in a Smart Classroom?

The equipment installed in most of the smart classrooms are:

- **Computer or Laptop:** A computer or laptop is a basic and necessary requirement of a smart classroom. Instead of writing on board with a chalk or marker, teaching process is done by presentation or images or multimedia in a smart classroom.
- **Projector:** it is an optical device. It projects stationery or moving objects onto a screen. They create an image by shining light through a transparent lens or by a laser.
- **Screen:** It is a surface used to display the images projected by a projector. The screens may be rigid wall-mounted screens, pull down screens, fixed frame screens, electrical screens, switchable projection screens, or mobile screens.

- **Microphone:** it is generally called as mic or mike. It is a device which converts audio signals into electrical signals. These signals are then transmitted, amplified or recorded.
- **Amplifier and Speaker:** These are the electronic devices which are used to increase the volume of a sound.
- **Podiums:** it is a platform usually made of wood which raises the person standing on it so that he becomes visual to all the audience.
- **Document Camera:** They are also called as visual presenters, visualizers, digital overheads. They are used for displaying an object to a large audience. A document camera magnifies and projects the images of a two dimensional as well as a three-dimensional object. The object is simply placed under the document camera. The camera takes its image and produces its live image by using a projector.
- **Smart Podiums:** They are also called smart boards or smart LCDs. It is an interactive pen display which can be connected to the computer or laptop externally via the USB port or RGB ports. It can be called as an external monitor with the facility of digital inking. With a smart podium, we can open documents, presentations, multimedia files and can write on them with a digital pen. We can also save our work.

Advantages of Smart Classrooms

There are many advantages of the smart classroom. Some of them are as follows:

- **Access to Online Resources:** In a smart classroom, the teacher can use the resources available on the internet for the better understanding of the students. For every subject, there are numerous resources available on the internet which can be accessed by the teachers as well as the students at any time in a smart classroom. Internet plays a major role in the learning process. Today's students are curious to know about everything. They want to gain extra knowledge about the ongoing topic in the class. This is a limitation of books. The books print only the syllabus. For those who want to learn beyond the syllabus, the internet is a very useful tool. Even for the syllabus, the resources available on the internet are very helpful.
- **Digital Medium for Notes:** In a regular classroom, the teacher writes on the board and the students note it down in their notebooks. In this methodology of teaching, the student's mind gets diverted in two directions, one is listening to the teacher and understanding and the other is writing or making notes for future reference. In this way, the students do not understand the topic properly or get confused or even sometimes, they do not listen to anything and just make

notes. This is a very dull approach in the teaching process. But this is not the case in a smart classroom. In a smart classroom, there is a digital approach for notes. The teacher, instead of writing on the board, teachers using PowerPoint presentation, word documents, images, videos, and audios. These resources are given to the students in pen drives, CDs, or they are emailed to them. In this way, the students do not have to write for making notes. Notes are provided to them and they can devote their full concentration to the lecture.

- **Advantageous for the Absentees:** If a student who studies in a regular classroom is absent for one day, it becomes difficult for him to gather all the notes of the classes he missed. Also, it becomes difficult for the teacher to explain the previous topics to the students who were absent. The students then take a shortcut method and get the notes photocopied from other students. In this way, the students do not understand anything from the classes he missed. In a smart classroom, the lectures are recorded. Whenever a student is absent he can anytime see the recorded lecture. These recorded lectures are also uploaded on the internet which can be accessed by the students anytime by a login id and password. The notes are also in digital form, so they can be easily provided to the student.
- **Ease of Understanding:** There are many topics in the curriculum which are very difficult to understand. They cannot be just taught by using a chalk and a black blackboard. They require an interactive teaching approach. With smart classes, the teacher can use multimedia to teach that topic to the students. The students learn more from what they see rather than from what they listen. So the students understand easily and effectively all the difficult topics in a smart classroom.
- **Makes Learning Enjoyable:** If we go and look in a regular classroom, the students are feeling very sleepy. Some of the students are not even listening to the teacher. They are busy talking with their friends. In short, the class becomes very boring for them. But it is not so in a smart classroom. The use of smart technology in the class makes the classroom a fun room. The class becomes interesting and enjoyable for the students. No student feels sleepy. It makes learning a fun process. The students who do not like to go to schools also start enjoying the school.
- **Improves the Academic Performance:** It is often seen that the students studying in a smart class get a better result than the students studying in a regular class. This is so because the understanding ability in the students studying in a smart classroom is way more than other students. The use of technology in the classroom for teaching increases the understanding of the students. The topic becomes clearer and the base of the subject becomes

stronger. Obviously, the students with a better hold of the subject and with strong basic knowledge of the subject will score more in exams.

- **Communication:** The technology used in a smart classroom makes video conferencing possible in the classroom itself. In this way, the students can communicate with the person who is an expert on the subject. In various schools and colleges with the facility of smart classrooms, guest lectures are organized weekly in which the students can communicate with the experts such as professors of IITs and IIMs or the people engaged in research work. This provides the students with extra information about their subjects which beyond the scope of the textbooks. The students get to know about the ongoing issues related to their subjects. They get acquainted with the upcoming technology and theories. This proves very beneficial for them in the near future.
- **Health Friendly:** There are some teachers who are very sensitive towards the use of chalk or markers. Some of them experience skin allergies when they write with chalk and some are allergic to the chalk dust. Some teachers are not comfortable with the smell of the ink used in the markers. For these teachers smart classrooms are advantageous. There is no or very less use of markers or chalk in a smart classroom. The teaching is done by using digital or e-resources. So, there is no chance of allergy.

Disadvantages of Smart Classrooms:

With so many advantages of the smart classrooms, there are a few disadvantages also which makes the use of smart technology quite limited. They are:

- **Costly:** The technology used in the smart classrooms is very expensive. All the schools cannot afford it. Even the schools which can afford it get over the budget after buying it. Some schools often take loans to buy the smart technology. Consequently, the fees of the schools are increased to meet the cost of the equipment. The admission fees of the schools with smart classrooms also hike. All parents cannot afford such an expensive education for their children. Every child cannot get access to the smart classrooms.
- **Skilled Faculty:** All teachers cannot use smart technology for teaching. The teachers have to be trained before they can start teaching in a smart classroom. The teachers are trained to use the technology in an effective manner. They are also trained to handle faults in the gadgets. This training also costs much. If the teachers are not properly trained the technology would become a disturbing factor rather than becoming helpful.
- **Maintenance:** Like all electronic gadgets, the gadgets used in smart classrooms require proper maintenance. The rooms in which these gadgets are installed

should be free from dust particles and should not face direct sunlight otherwise the gadgets will become faulty. Also, these equipment release a lot of heat in the surrounding so the rooms should be air-conditioned. Technical staff has to be appointed for the maintenance of devices. This further adds to the cost. The devices also need to be updated regularly which is a very time-consuming process.

- **Technical Faults:** All the electronic devices are prone to faults. If the equipment does not work one day, the day is considered as a waste. They require a lot of time to come back to their proper functioning state. No teaching can be done on that day. In this case, the teacher should always be ready for an alternative methodology for teaching that is, using the blackboard. But this would be a very dull experience for the teachers as well as the students who are regularly participating in smart classrooms. The workload of the teachers is also increased as they have to prepare the topic as per two different methodologies of teaching.

With advancement of technology such as streaming of video on the web, the need to have a classroom as a meeting point is no more. This is because smart classroom technologies have enabled listening of lectures from any place. Recording of what transpires in a lecture room has also avoided the problem of having to obtain the information together with the rest of the students as one can always listen at a convenient time and as many times as possible.

The rate of the student attendance is crucial as students will perform well in exams and practical lab if they go to classes consistently. Some of the students may not understand the importance of attending the class as they tend to skip the classes.

In University the attendance system is important as it will deter the rate of student skipping the class in which students are required to obtain at least 80% of the attendance in order to be eligible in taking the final exam of that particular subject. Thus, it is an obligation for lecturer to record student attendance in order to abide for this academic rule.

This is where autonomous attendance system comes in. The purpose of this project is to create a platform in terms of a smart classroom system that will bring accessibility to students and especially to lecturers. This autonomous

attendance system can be implemented in the university to track the percentage of attendance of each of the student automatically.

The data will be sent to the university server and be processed to ensure validity of the student in the correct class and time. Then the data will be visualized in terms of data and statistics in the website to be accessed by respective lecturer. Through the system, lecturer workload can be significantly reduced. At the same time, this system will encourage students to attend classes.

1.2 Attendance methodology

Attendance as we mentioned before for the students is an important task in class. When done manually it generally wastes a lot of productive time of the class. This proposed solution for the current problem is through automation of attendance system using face recognition. Face is the primary identification for any human. This project describes the method of detecting and recognizing the face in real-time using Raspberry Pi. This project describes an efficient algorithm using open source image processing framework known as OpenCV. Our approach has five modules – Face Detection, Face Preprocessing, Face Training, Face Recognition and Attendance Database. The face database is collected to recognize the faces of the students. The system is initially trained with the student's faces which is collectively known as student database. The system uses user friendly User interface to maximize the user experience while both training and testing which are collecting student images and taking attendance with the system. This project can be used for many other applications where face recognition can be used for authentication. Raspberry Pi usage helps in minimizing the cost of the product and the usability as it can be connected to any device to take the attendance.

Face acknowledgment can be utilized for validation. In this proposed system we take the attendance using face recognition which recognizes the face of each student.

There are five different levels of accesses which are the administrator, lecturer, student, university, administration and guest. Each of the users will have limited access and interfaces according to the user level as this is for security and privacy purposes.

The server will check the availability and validity of the student whether he or she is in the correct class, before the data is successfully saved in the MySQL database. If the attendance is successfully taken after checking all the information, a string of

a welcome message together with the student name ('Welcome <name of the student>') will be displayed on the LCD display. Through this, the student can know that their attendance is successfully taken. This is necessary as students might cheat their way to scan in a different class but in the same time to take their attendance.

1.3 Rapid growth of smart devices

With the rapid growth of smart devices and high speed networks, the Internet of Things (IoT) has gained wide acceptance and popularity as the main standard for low-power lossy networks (LLNs) having constrained resources. It represents a network where “things” or embedded devices having sensors are interconnected through a private or a public network. The devices in IoT can be controlled remotely to perform the desired functionality. The information sharing among the devices then takes place through the network which employs the standard protocols of communication. The smart connected devices or “things” range from simple wearable accessories to large machines, each containing sensor chips. For instance, the Lenovo smart shoes contain chips which provide support of tracking and analyzing fitness data. Similarly, the electrical appliances including washing machines, and refrigerators can be controlled remotely through IoT. The security cameras installed for surveillance of a location can be monitored remotely anywhere in the world. Apart from the personal use, IoT serves the community needs as well. Various smart devices which perform diverse functionalities such as monitoring surgery in hospitals, detecting weather conditions, providing tracking and connectivity in automobiles, and identification of animals using biochips are already serving the community specific needs. The data collected through these devices may be processed in real-time to improve efficiency of the entire system.

With the continuous advancements in technology a potential innovation, IoT is coming down the road which is burgeoning as an ubiquitous global computing network where everyone and everything will be connected to the Internet. IoT is continually evolving and is a hot research topic where opportunities are infinite. Imaginations are boundless which have put it on the verge of reshaping the current form of internet into a modified and integrated version. The number of devices availing internet services is increasing every day and having all of them connected by wire or wireless will put a powerful source of information at our finger tips. The concept of enabling interaction between intelligent machines is a cutting-edge

technology but the technologies composing the IoT are not something new for us. IoT, as you can guess by its name, is the approach of converging data obtained from different kinds of things to any virtual platform on existing Internet infrastructure. The concept of IoT dates back to 1982 when a modified coke machine was connected to the Internet which was able to report the drinks contained and that whether the drinks were cold. Later, in 1991, a contemporary vision of IoT in the form of ubiquitous

Smart classrooms have been successful due to the availability of aspects of interactive communication and collaboration instruments such as chat rooms, electronic forums like teleconferencing, and e-mail applications among others. Interaction in live and online classroom setting differ, with some situations being better handled electronically and others physically/live.

For this reason, it has not been possible to eliminate all aspects of traditional classroom setting for instance laboratories. Smart classrooms have changed people's perception about learning with most of them appreciating the fact that effective learning does not have to occur in a physical classroom as long as the objectives of learning are achieved.

We presented a new framework for recommending new pedagogical approaches for integrating IOT into smart classes and the concept of collaborative learning via IOT. Besides, the proposed approach IOT Learning cover other important dimensions, such as the impact of the IOT environment on the learners' level of knowledge and cognitive load. Future work aims to develop an intelligent environment based on an IOT authentication system that allows learners to identify themselves in intelligent classes.

This global analysis blends the opinions of those who have experienced the use of connected objects in class with those who have never done so. Also, there is the risk of confusing proven benefits with hopes of profits.

The overall responses of the 100 teachers are illustrated in the following graph.

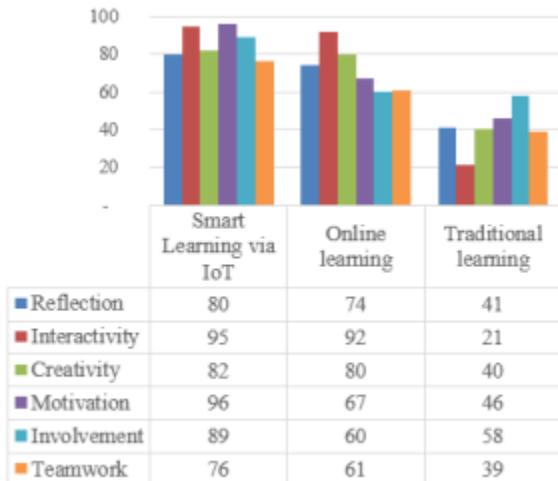


Figure 1-2. Comparison of different types of learning

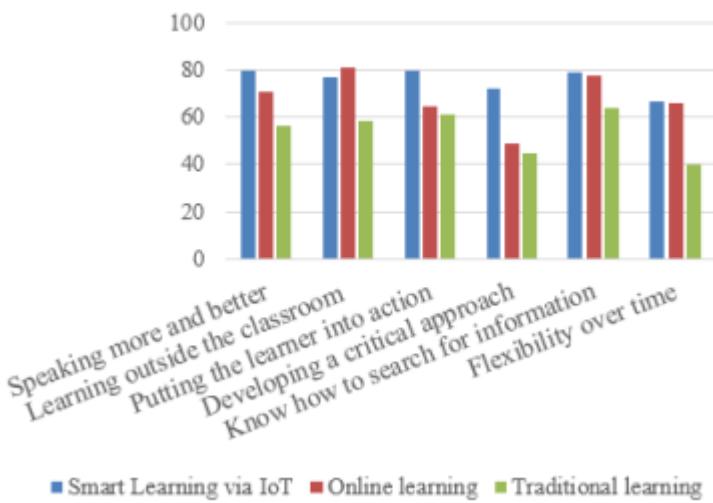


Figure 1-3. Advantage of Smart learning via IOT

The same project can be utilized for several security applications where authentication is needed to access the privileges of the respective system. It can be used in recognizing guilty parties involving in unauthorized business. Face recognition algorithm can be improved with respect to the utilization of resources so that the project can recognize more number of faces at a time which can make the system far better. Many variants of the project can be developed and utilized.

for home security and personal or organizational benefits. We can also trace a particular student in an organization quickly with the help of this system.

To conclude this section, it should be noted that the overall score for observing the benefits of connected objects is significantly influenced by seniority in the career. The youngest in the teaching profession have a much more positive opinion than those who have older

1.4 Conclusions

Nowadays, IoT is a smart network that connects all things to the Internet for exchanging information and communicating through exchange information and communicate through intelligent objects. There are enormous values in the adaptation of IoT throughout the education system. IoT enables the education system to be more relevant and effective. In the future, most educational institutions will gradually use IoT technologies to improve the learning process, and distance education. The new digital culture will create virtual schools that rely on Internet objects to encourage off campus learning and expand participation in modern learning.

2. Chapter Two: IOT

2.1 Introduction.

Using the Internet to both receive data from and/or provide control commands to devices. The “Internet of Things” (IoT) is a phrase that was first used in 1999 by Kevin Ashton while he was working at MIT’s Media Center. He meant it to represent the concept of computers and machines with sensors, which connect to the Internet to report status and accept control commands.

With the rapid development of Internet of Things (IoT), it has now become a buzzword for everyone who works in this area of research. Further, it is seen that with the rapid development of sensors and devices with their connection to IoT become a treasure trove for big data analytics. It has found numerous applications in developing smart cities where predictions of accidents and traffic flow in the cities can be effectively monitored; smart health care where the doctor is able to get useful information from the implant sensor chip in the patient’s body; industrial production can also be enhanced manifolds by efficient prediction of the working of machinery and smart metering in helping the electric distribution company to understand the individual household energy expenses and making smart homes with connected appliances to name a few. The 21st century is for IoT, where it is viewed as a network of physical devices coming together from electronics, sensors, and software. It is envisioned that the network of approximately 27 billion of physical devices on IoT are presently available and the list grows. These devices (Cars, Refrigerators, TVs etc.) can be uniquely identifiable through embedded computing system and can be connected from anywhere through suitable information and communication technology, to achieve greater service and value. The “THING” in IoT means everything and anything around us that includes machines, buildings, devices, animals, human beings, etc. Today’s, smart health care, smart homes, smart traffic, and smart household devices use this technology for a better digital world.

The IoT, in reality, has been around for a long time, but it didn’t have a name. Machine-to-machine (M2M) communications has been in existence for many decades, often using dedicated networks that eventually converged over to the Internet.

Note that IoT(Internet of things) differs from IoE (Internet of everything) and let’s take little review about IoE.

2.2 Internet of Everything (IoE):

The Internet of Everything (IoE) is a concept that extends the Internet of Things ([IoT](#)) emphasis on machine-to-machine ([M2M](#)) communications to describe a more complex system that also encompasses people and processes.

The concept of the Internet of Everything originated at Cisco, who defines IoE as "the intelligent connection of people, process, data and things." Because in the Internet of Things, all communications are between machines, IoT and M2M are sometimes considered synonymous. The more expansive IoE concept includes, besides M2M communications, machine-to-people (M2P) and technology-assisted people-to-people (P2P) interactions.

The Internet of Things, in its broadest conceptualization, includes any type of physical or virtual object or entity that can be made addressable and given the ability to transmit data without human-to-machine input -- those are the [things](#) in the IoT. Things are often items that would not have been networked in the past; automation of thing communications is also central to the IoT concept. The IoE, on the other hand, also includes user-generated communications and interactions associated with the global entirety of networked devices.

Table 2-1. IoT vs IoE

INTERNET OF THINGS VERSUS INTERNET OF EVERYTHING	
INTERNET OF THINGS	INTERNET OF EVERYTHING
Network of physical devices and items embedded with electronics to enable connectivity and to exchange data	Border concept than Internet of Things which defines as the intelligent connection of people, processes, data and things
Communication occurs between Machine to Machine	Communication occurs between Machine to Machine (M2M), Machine to People (M2P) and technology assisted people to people (P2P)
IoT is less complex than IoE	IoE is advanced than IoT Visit www.PEDIAA.com

2.3 IoT value in different settings.

The Internet of Things has the potential to fundamentally shift the way we interact with our surroundings. The ability to monitor and manage objects in the physical world electronically makes it possible to bring data-driven decision making to new realms of human activity—to optimize the performance of systems and processes, save time for people and businesses, and improve quality of life .From monitoring

machines on the factory floor to tracking the progress of ships at sea, sensors can help companies get far more out of their physical assets—improving the performance of machines, extending their lives, and learning how they could be redesigned to do even more. With wearable devices and portable monitors, the Internet of Things has the potential to dramatically improve health outcomes, particularly in the treatment of chronic diseases such as diabetes that now take an enormous human and economic toll.



Figure 2-1. value potential of the Internet of Things

1. HUMAN

We focus here on IoT applications in the context of the human body as the setting in which IoT systems are deployed. These applications fall into two broad categories: improving health and raising productivity. Unlike other IoT applications, where a reading from a sensor might initiate a specific action—turning off a valve, for example—in the human setting, sensor data provide information that people will use to guide their actions and decisions. The adoption of IoT applications in health and fitness is well under way. Based on current usage and likely growth rates, we project that the economic impact of IoT in human health and wellness could be \$170 billion to nearly

\$1.6 trillion globally in 2025. The largest source of value would be using IoT devices to monitor and treat illness (\$170 billion to \$1.1 trillion per year). Value would arise from improving quality of life and extending healthy life spans for patients with chronic illnesses, and reducing cost of treatment. The second-largest source of value for humans would be improved wellness—using data generated by fitness bands or other wearables to track and modify diet and exercise routines. Emerging applications have the potential to transform a wide range of health-care therapies. implantables, ingestibles, and injectables, such as nanobots have the potential eventually to replace many surgeries with less invasive procedures that could offer faster recovery, reduced risk of complications, and lower cost. While these technologies are still in development, if they are adopted widely in the next ten years, they have the potential to raise the economic impact of IoT in health care substantially beyond the \$1.6 trillion we estimate here. Another set of human applications aimed at raising workplace productivity is also emerging. We describe these applications in the human setting, but we assign their potential economic impact to settings where they will be used, such as offices, retail environments, and factories. For human IoT applications to generate the maximum benefits in health will require improvements in cost, new behaviors by health-care providers and payors, and advances in analytics. Among necessary changes, some of the most important will involve how health care is delivered (using devices and analytics rather than human caregivers) and reimbursed. Payors (commercial insurers and government-run systems) will need to be convinced that investments in IoT systems are justified, especially in cases where IoT improves outcomes but also increases treatment cost. It will also need to be shown that IoT solutions can have a real impact on one of the most vexing problems in health car today: human behavior. Using these systems to convince healthy people to change their living habits and to help sick patients adhere to doctors' prescriptions would be a true breakthrough. In addition to wearables and home health monitors, IoT devices for human health include implantables, ingestibles, and injectables, such as nanobots that can clear arteries or help detect early-stage cancer. These devices have not yet reached the clinical trial stage, and we do not attempt to size their potential impact in 2025. However, when

they are ready for widespread adoption, their impact could be as large as or larger than the benefits of the other technologies we discuss here.



Figure 2-2.iot-healthcare-wearable-heart-rate-tracker

Potential economic impact

Overall, we estimate that the use of IoT technologies in human health applications could have an economic impact of \$171 billion to \$1.6 trillion globally in 2025. The largest source of impact would be in treating patients with chronic diseases, which could be worth nearly \$1.1 trillion per year globally. This is based on two sources of value—cost savings in treatment and the value of longer lives and improved quality of life that patients with chronic conditions could enjoy if IoT monitoring helps them avoid disease complications. We estimate that cost savings have a value of \$110 billion to \$470 billion per year in 2025, assuming savings of 10 to 15 percent in advanced and developing economies. The larger source of value could be improvements in life span and quality of life, which could have a value of up to \$520 billion per year globally. About 75 percent of this impact would be in advanced economies, where the costs associated with treating these conditions are higher and the potential economic benefits of extended and improved quality of life are greatest.

Monitoring and treating illness

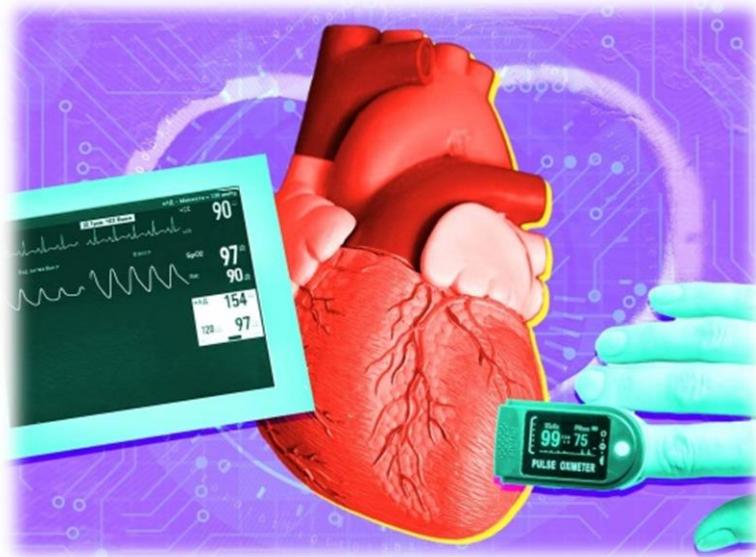


Figure 2-3.real-time-for-early-detection-of-heart-diseases

Using IoT technology for more continuous and consistent monitoring of patients with chronic diseases can help the patients avoid medical crises, the hospitalizations, and the complications. Monitoring with conventional tools has fewer benefits for four reasons: 1)they provide only episodic readings (when blood is drawn, for example); 2) tracking must be done in high-cost settings such as hospitals, which leads to treatment avoidance; 3) patients often fail to adhere to prescribed treatment; and 4) a limited ability to identify problems in a timely manner before they develop into costly or even fatal conditions.

2. HOME

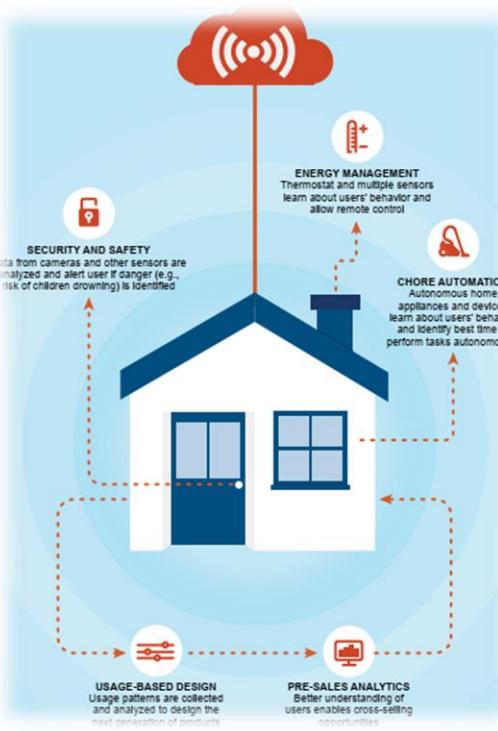


Figure 2-4. Various IoT applications in homes

In the home setting, we assess the impact of Internet of Things applications relating to the operation of homes, as energy management, security, and automation of domestic chores. We do not include human health and fitness applications (wearable fitness monitors), even though they are commonly used in the home. Those uses are covered in the human setting. We estimate that IoT applications in the home could have an economic impact of as much as \$350 billion per year. The potential economic impact in the home setting is less than in settings such as factories, but it could change how consumers interact with their surroundings and spend their time at home. By far the largest opportunity in the home setting is in automating domestic chores. This work is not counted in national productivity data, but has an enormous impact on how people spend time and money. In the United States alone, household activities (cleaning, washing, preparing food, gardening, caring for pets, and so on) and purchasing home goods and services require 230 billion labor hours per year. Globally we estimate the value of time spent on domestic chores will be more than \$23 trillion in 2025. We also estimate that devices such as self-guided vacuum cleaners and lawn mowers

can cut the time required for household activities by 17 percent.

Potential economic impact

In the home, the Internet of Things has the potential to create an economic impact of \$200 billion to \$350 billion annually in 2025. Promising uses are chore automation, energy management, safety and security, usage-based design, pre-sales analytics, and personal loans.

Energy management

Using sensors and predictive algorithms, smart thermostats can detect when no one is home and adjust the temperature to conserve energy. Over time the smart thermostat could learn about usage patterns and adjust heating or cooling to have the home at the right temperature when residents are due home. Connected washers and dryers (working with smart meters installed by utility companies) could get information about energy prices to delay cycles during peak energy consumption periods. IoT-enabled energy management applications could have an economic impact of \$50 billion to \$110 billion globally in 2025 through savings on heating, air conditioning, and overall electricity use. IoT devices could help reduce electricity bills by ensuring that devices are powered on only when necessary and by reducing usage when energy is most expensive. Nest claims that its smart thermostats save 20 percent off heating and air-conditioning bills by turning on these systems only when occupants are expected to be home. Additional savings are possible through the use of smart meters and smart appliances, which would allow automatic shutdown of appliances during times of peak electricity demand. We estimate that adoption rates for IoT energy-control applications could reach 25 to 50 percent in advanced economies in 2025 and 4 to 13 percent in developing economies. The US Energy Information Administration estimates that 37 percent of current US residences have programmable thermostats to control heating and that 29 percent have programmable devices for running cooling systems. Given the competition in this market, we expect smart energy control devices to come down to price points where owners of programmable thermostats will convert to IoT-enabled devices. We would also expect more consumers to seek energy-conservation tools.

Safety and security

IoT sensors and systems can greatly reduce losses to consumers from break-ins, fire, water leaks, and injuries in the home. Combining sensors, cameras, and

powerful analytics, future IoT systems could sense when inhabitants are at risk and issue alerts to fire, police, or emergency services for prompt action. For example, cameras and sensors could be installed near pools so that parents are alerted immediately if children are in danger. Based on expert interviews, we estimate that willingness to pay for such security systems could be as much as \$400 per year per household. We have used a more conservative estimate of \$180 per year per household, to account for the higher adoption rates we would anticipate as prices fall. Through early detection, sensors could also help reduce property damage from water leaks and fire. Sensors to detect home leaks are already being sold. Use of such IoT systems could help reduce home insurance premiums by up to 10 percent, we estimate. In total, we estimate that the economic benefit of IoT-based safety and security systems could be \$15 billion to \$20 billion per year. We estimate that adoption rates for safety and security devices could be 18 to 29 percent in advanced economies in 2025 and 9 to 13 percent in developing economies.

3. RETAIL ENVIRONMENTS

We define retail environments broadly as physical spaces where consumers engage in commerce—considering or purchasing goods or services. This includes traditional stores, such as department stores and grocery stores, as well as showrooms where goods are on display but not available for sale. It also includes physical spaces where services are purchased, such as bank branches, theaters, and sports arenas. Our analysis covers only physical environments where IoT technologies can be deployed, not online retailing. Retail environments have undergone significant change over the past two decades due to the introduction of information technologies, including the rise of online shopping. The Internet of Things has the potential to cause even greater disruption, but IoT can also provide traditional retailers with the tools to compete—and coexist—with the online retail world as “omni-channel” shopping erases the distinction between online and offline shops. The Internet of Things, for example, can guide the shopper to the item she has been looking at online when she enters the store and text her a personalized coupon to make the purchase in-store that day. IoT technology can also provide data to optimize store layouts, enable fully automated checkout, and fine-tune inventory management. These and other innovations could enable new business models and allow retailers to improve productivity, reduce costs, and raise sales.

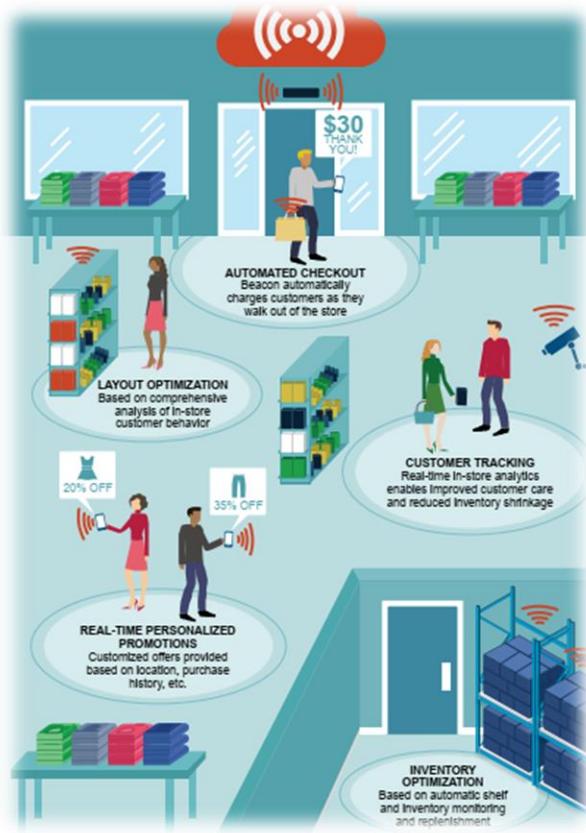


Figure 2-5.Examples of IoT uses in retail

IoT adoption in the retail setting will depend not only on the evolution of technology (lowercost sensors, for example) but also the development of new business processes. IoT systems require modern store formats and investments in data systems and electronic payment systems. This is not an issue in advanced economies, but it could hold back IoT adoption in developing economies. Tiny independent “mom and pop” shops account for the majority of retail trade in places such as Mexico and India and provide employment for millions of low-skill workers. Some countries have adopted policies to protect these players from more efficient modern stores. Widespread IoT adoption would affect players across the value chain, including employees and consumers. It has the potential to reduce the need for labor on the selling floor and at checkout, while raising the amount of revenue per customer (increasing the “shopping basket”) through customization and cross-channel (online/offline) selling. Consumers would gain more value through convenience, time saving, and more attractive customized promotions. To

remain competitive, companies would need to master new ways of operating and learn to collaborate closely with technology and data vendors. Historically, many retail companies have been slow to adopt technology due to industry fragmentation, lack of scale, and limited margins in the industry. In recent years, however, adoption of payments, security, and inventory control systems has accelerated, even among smaller companies (in advanced economies). By adopting Internet of Things technologies, retailers can improve their economics by reducing shrinkage (losses due to theft by customers and employees), lowering inventory costs, raising productivity, and improving the customer experience.

Real-time in-store promotions: With beacons that connect to mobile phones to track customers within the store, retailers can launch custom promotions in real time. Once the customer is identified by his or her phone, algorithms can combine historical information about the customer's preferences and lifestyle with current in-store location data to create unique offers. Over time, these systems can develop customer profiles that include not only data about what they have purchased, but also what they are willing to pay. Real-time advertisements and promotions based on this information can increase spending per customer, giving the advertiser a higher return on investment and raising productivity. In a theater or sports arena, patrons selected by particular criteria (such as frequent attendance) could be offered last-minute upgrades to unsold premium seats at discount prices. We estimate that real-time personalized promotions can increase productivity in retail environments by 3 to 5 percent, leading to a potential economic benefit of \$89 billion to \$348 billion a year. These techniques are likely to have the greatest benefit for incumbents, which are likely to have more data on their customers. However, data aggregators and other third parties can enable newer entrants to develop real-time promotion capabilities. Real-time personalization could be used in banks for cross-selling and in theaters, hotels, and restaurants for upgrades or other loyalty promotions.

Automated checkout

Checkout is one of the most labor-intensive processes in retail and a frequent source of frustration for customers, who must wait in line and go through complex, multistep transactions to make payments. While self-checkout systems have been introduced in some retail environments, most offer only

limited improvement over the traditional cashier system and its card- or cash-based transaction process. The Internet of Things has the potential to completely automate checkout by scanning the contents of shopping carts and automatically charging the sale to the customer's mobile payments account, allowing a consumer to walk out of a store without pausing. The system would read the electronic tags on the items in the cart and a checkout system would add up the prices of the items and relay the information to a wireless payment system that would debit the customer's smartphone as it passes. This would lead to lower costs for the store as well as time savings for the consumer. We estimate that automated checkout could reduce cashier staff requirements by up to 75 percent, resulting in savings of \$150 billion to \$380 billion a year in 2025. It would also reduce checkout queue times by 40 to 80 percent, providing a potential economic benefit of \$30 billion to \$135 billion of saved time in 2025. For automated checkout to be used across all types of retail stores, lower-cost electronic tags will be needed.

4. OFFICES

We define offices as commercial spaces where knowledge workers work. Internet of Things technology has applications in the office setting that are similar to those in the home setting—principally managing energy and security systems. Here we exclude retail spaces and the care-providing spaces in hospitals, which are usually considered to be commercial spaces, but are covered, in our analysis of the retail environments and factories settings, respectively.

Potential economic impact

It is estimated that use of IoT technologies to manage office spaces could have an economic impact of \$70 billion to \$150 billion per year in 2025.

Energy and environment management

Commercial spaces account for 20 percent of energy consumption in advanced economies such as the United States and approximately one-sixth of energy consumption in developing economies.³⁴ Around half of this energy is consumed in office spaces and much of it is wasted by heating, cooling, and lighting unoccupied offices. Intelligent energy management systems can be used to automatically sense when a room is unoccupied or occupied and adjust heating or cooling and lighting as needed. It is estimated that intelligent energy management systems can reduce energy use in offices by 20 percent, producing a potential economic impact of \$11.7 billion to \$20.5 billion per year in 2025.

What is the common thread that binds them together? And why the name?

All the cases we saw used the *Internet* to send, receive, or communicate information. And in each case, the gadget that was connected to the Internet wasn't a computer, tablet, or mobile phone but an object, a *Thing*. These Things are designed for a purpose: the umbrella has a retractable canopy and a handle to hold it. A bus display has to be readable to public transport users, including the elderly and partially sighted and be able to survive poor weather conditions and the risk of vandalism. The sports bracelet is easy to wear while running, has a display that is large enough and bright enough to read even when you are moving, and will survive heat, cold, sweat, and rain. Many of the use cases could be fulfilled, and often are, by general-purpose computers. Although we don't carry a desktop PC around with us, many people do carry a laptop or tablet. More to the point, in almost every country now, most people do carry a mobile phone, and in many cases this is a smartphone that easily has enough power for any task one could throw at a computer. Let's see how well one could replicate these tasks with a smartphone. Viewing your bus provider's timetable with a smartphone web browser seems to fulfil the same function at first glance. But just consider that last phrase, "at first glance". On arriving at the bus stop, one can simply glance at the computerized timetable and see when the next bus is due. With a smartphone, if you have one and can afford the data use (which may be prohibitive if you are a foreign tourist), you have to take the phone out of your pocket or bag, unlock it, navigate to the right website (this may be the slowest and most complicated part of the process, whether you have to type the URL or use a QR code), and read the data from a small screen. In this time, you are not able to fully concentrate on the arriving buses and might even miss yours. You can track your runs with an app on your smartphone, and many people do: the phone has GPS, many other useful sensors, processing power, an Internet connection, and a great screen.

2.4 Project interest

Our focus is on creating useful project that take advantage of the tremendous communication capabilities provided by an Internet connection.

We should note that the Arduino platform uses three slightly different implementations for Internet connectivity, which I classified as one platform. We can compare briefly between different platforms was a deliberate and purposeful decision on our part to show you what is involved in creating projects using different development infrastructures yet still establishing a working Internet

connection. These three hardware and software development platforms are listed in Table 1

Hardware Platform	OS	Language(s)
Raspberry Pi	Linux-Raspbian (Debian) distribution	Python, java
Arduino	WIN7-for development	.NET framework processing
Beaglebone Black	Linux-Wheezy (Debian) distribution	Python

Table 2-2. IOT Software development Platforms

Creating a project that is equipped with sensors and is capable of both sending and receiving data via the Internet is a bit challenging, especially to those readers who are attempting to do so for the first time. Let's start this journey with a discussion of hardware as that seems easiest for most folks to handle and is absolutely required for these projects .This project aims to make it easier at the learning environment to save time using one of the IoT platforms like Raspberry Pi It is often seen that the students studying in a smart class get a better result than the students studying in a regular class. This is so because the understanding ability in the students studying in a smart classroom is way more than other students. The use of technology in the classroom for teaching increases the understanding of the students. The topic becomes clearer and the base of the subject becomes stronger. Obviously, the students with a better hold of the subject and with strong basic knowledge of the subject will score more in exams because of the advantage of saving time and focus their work on the target which take result of reclaiming more information and due to improving that will be happen from feedback results and voting and reducing cheating at quiz and exams. We will mention our platform we used in this project which is Raspberry Pi model in detail in the next chapter.

2.5 Conclusion

Internet of Things offers some pretty **interesting applications** in making our lives easier like in Healthcare, Transportation, and Agriculture. However, various factors like security, privacy and data storage also need to be considered.

It is also worth noting that **things have been connected to networks for ages** without the guise of “Internet of Things”.

3. Chapter Three: Raspberry pi

3.1 Introduction

The Raspberry Pi is a credit card-sized computer designed and manufactured in the UK with the initial intention of providing a cheap computing device for education. Since its release, however, it has grown far beyond the sphere of academia.

Its origins can be found in the University of Cambridge's Computer Laboratory in 2006. Computer scientist Eben Upton, along with Rob Mullins, Jack Lang and Alan Mycroft, were concerned that incoming computing undergraduate students had grown divorced from the technical aspects of computing. This was largely due to school syllabuses that placed an emphasis on using computers rather than understanding them.

Off the back of this initial concern, the Raspberry Pi foundation was formed. Over the next six years the team worked on developing a cheap and accessible device that would help schools to teach concepts such as programming, thus bringing students closer to understanding how computing works.

The Raspberry Pi's initial commercial release was in February 2012. Since then, the board has gone through a number of revisions and has been available in two models, those being Model A and Model B.

The Model A device is the cheaper and simpler of the two computers and the Model B the more powerful, including support for Ethernet connectivity.

In February 2015, the Raspberry Pi 2 Model B was released, and this is the device discussed in this book.

The new Raspberry Pi 2 is significantly more powerful than previous versions, opening us up to many new possibilities.

We will now look at the hardware of the device to get a basic understanding of what it is capable of doing. Future chapters will build upon the basics presented here.

3.2 Raspberry Pi hardware specifications

The new Raspberry Pi is built on the back of the Broadcom BCM2836. The BCM2836 is a system-on-a-chip processor containing four ARM cores and Broadcom's Video Core® IV graphics stack.

In contrast to this, previous Raspberry Pi A and B models only contained a single core.

On top of this, several other components make up the device, including USB, RCA, and microSD card storage. The previous Raspberry Pi Model B only contained two USB drives and a micro USB compared to the four USB drives and micro USB of the second version.

You can read a good breakdown of how the two boards standup to each other by visiting the following website: <http://www.alphr.com/raspberry-pi-2/1000353/raspberry-pi-2-vs-raspberry-pi-b-a-raspberry-pi-comparison>.

So, compared to earlier models, version 2 is a far more capable computer, yet still remains at the same price. The added benefit of having multiple cores allows us to explore different programming techniques for utilizing them.

Next, we shall cover the core components of the Raspberry Pi board in more detail. The following is an image of the board with a description of each component:



Figure 3-1. Raspberry Pi 2 module

Dimensions

The Raspberry Pi 2 is a small machine measuring only 85.60 mm x 56 mm x 21 mm and weighing approximately 45g. This small size makes it suitable for embedded projects, home automation devices, arcade machines, or building small multi-device clusters.

System on Chip

The System on Chip (SoC) architecture that the Raspberry Pi 2 implements is the Broadcom BCM2836, which we touched upon earlier in this chapter. This contains a CPU, GPU, SDRAM, and single USB port. Each of these items is discussed in more detail under the appropriate heading.

CPU

A central processing unit is the brain of your Raspberry Pi. It is responsible for processing machine instructions, which are the result of your compiled programs.

The BCM2836 implements a 900 MHz quad-core ARM Cortex-A7 processor. This runs on the ARMv7 instruction set.

GPU

The graphics processing unit (GPU) is a specialist chip designed to handle the complex mathematics required to render graphics.

The Broadcom VideoCore Iv 250 MHz supports OpenGL ES 2.0 (24 GFLOPS) Mpeg-2 and VC-1 (with license). It also includes a 1080p30 H.264/MPEG-4 AVC decoded/encoder.

SDRAM

The Raspberry Pi 2 comes equipped with 1 GB of SDRAM, which is shared between the GPU and CPU.

4 USB 2.0 ports and 1 SoC on-board USB

The previous version of the Raspberry Pi Model B contained only a single micro USB port and a two standard USB ports. The Raspberry Pi 2 has been expanded to include an onboard 5-port USB hub.

This allows you to connect four standard USB cables to the device and a single micro USB cable. The micro USB port can be used to power your Raspberry Pi 2.

MicroSD card port

The microSD card is the main boot and storage mechanism of the Raspberry Pi. It is upon the microSD card that you will load your operating system and store data. Later in this book we will look at using the microSD purely for booting the Raspberry Pi, and then using a USB hard drive as a storage mechanism. In this chapter, we will delve into how we can setup the SD card with the Raspbian operating system.

Ethernet port

One of the benefits of the Raspberry Pi 2 Model B is that it contains an Ethernet port. Many Raspberry Pi packages available on Amazon and similar stores include a wireless USB dongle; however, this results in you having to use up a USB port. If you plan to place your Raspberry Pi near a router or switch or have enough Ethernet cable, then you can connect your Raspberry Pi directly with the Ethernet jack.

The Raspberry Pi 2 supports 10/100 Mbps Ethernet, and the USB adapter in the third/fourth port of USB hub can also be used for Ethernet via a USB to Ethernet adapter.

Ethernet to USB adapters can be purchased from most good electronics stores and you can read more about the technology at.

Audio

The Raspberry Pi 2 implements the Inter-IC Sound (I2S) serial bus for audio input and output. This allows the device to connect multiple digital audio devices together. A 3.5mm TRRS jack is available and shared with the analog video output. The HDMI component also provides digital audio output.

GPIO pins

The main method for interacting with electronic components and expansion boards is through the general purpose input/output (GPIO) pins on the Raspberry Pi.

The Raspberry Pi 2 Model B contains 40 pins in total. Future chapters will also look at how we can program these to control electronic devices.

As the acronym suggests the GPIO pins can accept both input and output commands and can be controlled by programs in a variety of languages running on the Raspberry Pi.

The input for example could be readings from a temperature sensor, and the output a command to another device to switch an LED on or off.

Video – analog TV out

As well as providing a digital method for hooking up to a TV or monitor, the Raspberry Pi 2 also comes with analog support. The method of connection is commonly known as a composite or RCA port and earlier models of the Raspberry Pi came specifically with an RCA jack. RCA cables typically come with three connectors, two for audio and one (often yellow) for video.

With the release of the Raspberry Pi 2 the composite video (RCA) and 3.5 mm audio jacks functionality has been merged into a single TRRS hardware component. Therefore, if you wish to use video through this port, you may need to get a 3.5mm Mini AV TRRS to RCA cable instead. These can be found at any good electronics stores or on Amazon.

The Raspberry Pi 2 supports both PAL and NTSC standards.

Video – HDMI port

Also included is a High-Definition Multimedia Interface (HDMI) port. This allows the Raspberry Pi 2 to be hooked up to high definition devices such as televisions and monitors. This port provides a digital alternative to the TRRS jack.

Let's review a comparison between PC and Raspberry Pi

Table 3-1. Comparison between PC and Rasberry Pi

	Raspberry Pi	Computer
Construction	It is a motherboard. All components are soldered directly onto the motherboard	The basis is the motherboard. It has other components such as RAM, Storage, CPU, and GPU

		that are connected to the motherboard using standard connectors.
Size	Average 85.60 mm × 56.5 mm × 17 mm	14 × 7 inches for the mini-tower and 24 × 8 inches for the full tower cases
Price	\$25 for model A and \$35 for model B motherboard.	\$400-\$3500 for the entire computer.
Memory and Storage	RAM between 1 and 8 GB. Micro SD card for storage.	Average RAM of 4GB and Hard Drive/SSD for storage.
Connectivity	Has 4 USB ports, 1 HDMI monitor minimum, 1 audio port, an Ethernet port. Has Bluetooth and Wi-Fi capability. No Microphone, no additional ports.	Highly depends on the computer model, but usually includes at least the equivalent, with options for easy expanding.
Architecture	ARM	AMD64
Screen	No screen	Can have a screen

➤ Basic hardware needed

In order to get up and running with your Raspberry Pi 2 you will need the following additional hardware components:

- MicroSD card
- Micro USB power cable

- Monitor—preferably HDMI
- HDMI cable or 3.5mm to RCA AV cable
- USB keyboard
- USB mouse
- Protective case—optional
- Wi-Fi dongle or Ethernet cable

Before we can power up and start using our Raspberry Pi, however, we need to install an operating system on a microSD card.

➤ **The microSD card – the main storage and boot device of the Raspberry Pi 2**

A micro secure digital (microSD) card is a portable high performance storage medium used in a variety of electronic devices including cameras, phones and computers. You may already be familiar with them if you use one of the devices we have just listed.

Our Raspberry Pi 2 comes equipped with a microSD slot, which lets us use a microSD card as our main storage and boot mechanism. The card is therefore used in a similar manner to a hard drive on a traditional computer or portable device.

The previous Raspberry Pi models used a standard SD card, which was much larger. Therefore, the microSD card saves space on the circuit board and does not poke out as far, reducing the risk of it being broken.

When choosing a microSD card for your projects, there are a variety of brands on the market, and they come in a range of storage sizes running into the tens of gigabytes.

For the projects in this book we recommend using a card with a large amount of storage and you should look at choosing a card that is at least 8 GB in size. The NOOBS application, for example, requires a card of at least this size.

We will now discuss the option of purchasing a microSD card preinstalled with the Raspbian operating system or New Out Of the Box Software(NOOBS) versus formatting and installing the operating system ourselves.

For those who really wish to understand the Raspberry Pi 2 in detail, installing the operating system from scratch may be a more rewarding experience.

Preinstalled microSD card versus creating your own

A number of websites offer microSD cards preloaded with one of the operating systems that are available for the Raspberry Pi 2.

These are a good solution for anybody looking to get up and running quickly or who are not comfortable installing an operating system by themselves from scratch. They are also useful for those who do not have second computer to work with in order to format a new microSD card.

The official Raspberry Pi distributions Element 14 also offer a preinstalled microSD card equipped with NOOBS, a Raspberry Pi 2 operating system boot loader.

It should be noted that if you do not have a home Mac or PC accessible to format a new blank microSD card, then we would recommend acquiring a preformatted card. This should come loaded with either Debian Jessie Raspbian, or the NOOBS boot loader application.

The NOOBS operating system installation manager

This book assumes that the reader will be installing the Raspbian operating system himself or herself. The simplest method for doing this is to install the NOOBS operating system installation manager onto your microSD card.

NOOBS makes the setup of your Raspberry Pi 2 easy and also provides you with a mechanism for choosing other operating systems that are compatible with the Raspberry Pi.

When installing Raspbian for the first time via NOOBS you will also be presented with the raspi-config screen. This provides some handy shortcuts that allow you to do the following:

- Expand the file system
- Change the user password
- Enable boot to desktop
- Change language
- Enable the camera if you have purchased the peripheral
- Add to Rastrack Raspberry Pi Map
- Overclock your Raspberry Pi
- Explore some advanced configuration options

If you choose not to install Raspbian via NOOBS, then the following section will guide you through the process. If you are using NOOBS you can skip to the Raspbian installation wrap-up section.

Downloading the latest version of Raspbian

Once you have obtained a copy of the operating system you can move onto formatting your microSD card and installing the image.

The HDMI port is ideal for streaming video and audio to your TV or monitor.

➤ **Setting up your microSD card and installing the Raspbian operating system**

The Raspbian installation process involves two steps:

Formatting the microSD card to the FAT file system

Copying the Raspbian image to the card

It is important that we quickly look at what File Allocation Table (FAT) is and why we need it.

FAT is a method for defining which sectors of a disk or microSD card files are stored in and which sectors on the disk are free to have new data written to them.

The standard has its origins in the 1970s for use on floppy disks and was developed by Bill Gates and Marc McDonald.

Due to its simplicity of implementation and robustness, this standard is still used on SD and microSD cards today. Therefore, it is the format you will need in order to install the Raspberry Pi's operating system onto your microSD card.

Due to its widespread adoption you may find a microSD card you purchase is already formatted to FAT.

We recommend, however, formatting any new cards you purchase to ensure you do not encounter any problems.

The official Raspberry Pi website provides handy how-to guides for the three major operating systems on how to format and install the Raspbian image.

The following are guides to formatting an SD card for your particular operating system:

Windows (<https://www.raspberrypi.org/documentation/installation/installing-images/windows.md>)

Mac OS X (<https://www.raspberrypi.org/documentation/installation/installing-images/mac.md>)

Linux (<https://www.raspberrypi.org/documentation/installation/installing-images/linux.md>)

Having completed installing the operating system we can now look at some final configuration before exploring some interesting features of Raspbian.

Raspbian installation wrap-up

The following section assumes you have your Raspberry Pi connected to a monitor and with a keyboard and mouse available. It also assumes you have your configuration set to boot to desktop and have powered up and logged into your device.

You should at this point connect your device to your home router. If you are planning on using Wi-Fi, read on.

Now that you have successfully installed Raspbian you should see the Linux desktop

Tip

If you do not see the desktop, but the command line instead, you can type startx to start the GUI.

This desktop contains icons in the top menu linking to a number of programs installed by default with the operating system.

One important icon is the link to LX Terminal. This icon launches the Linux terminal window. Click on this icon and you should see the command line load.

The following tasks in this section can all be performed in this window.

As a handy shortcut you can also load the raspi-config application at any time by typing the following command:

- sudo raspi-config

Tip

You can read about the sudo command here: <https://www.sudo.ws/>.

If you update settings in this manner you may need to reboot the Raspberry Pi for them to take affect.

Check SSH is running

In order to connect to our Raspberry Pi 2 from another device via a terminal window we need to ensure that the Secure Shell(SSH) server is up and running. SSH is the default mechanism for secure communication between our Linux machines. If you used NOOBS to install the OS you may have configured the SSH server at this point via the advanced options. We can check that the SSH service is running successfully as follows.

Open up a terminal window from the Raspbian desktop and type the following command:

- ps aux | grep sshd

The following sshd process should be displayed. This tells us the services are up and running:

- root 2017 0.0 0.3 6228 2892 ? Ss 15:13 0:00 /usr/sbin/sshd

If the SSH process does not appear, it is simple to start it. Enter the following command into the terminal:

- sudo /etc/init.d/ssh start

After you have executed this command try running the following again and check that the sshd process is now running:

- ps aux | grep sshd

By default, to login to the Raspberry Pi 2 over SSH you will be prompted for a username and password. If you have not changed this the username is pi and the password is raspberry.

In addition to the username and password method, we can also use an RSA key to authenticate and gain access to the Raspberry Pi over a network. We discuss this process next.

RSA key generation for SSH

RSA keys are a useful method to login to the Raspberry Pi. They remove the need to enter a username and password and lock down access to a handful of hardware devices. A RSA key consists of two parts: a public and private key. A public key can be shared with anyone and any machine with that key on and can in theory let you have access. Therefore, if you purchase more Raspberry Pis, you can place your public key on each, and negate the need to remember multiple passwords.

The private key portion should be kept secret and is located on the machine you will use to access your Raspberry Pi from. For security reasons it is best to keep the private key on a single device.

You can read more about RSA keys and the cryptographic theory behind them here: [https://en.wikipedia.org/wiki/RSA_\(cryptosystem\)](https://en.wikipedia.org/wiki/RSA_(cryptosystem)).

One important piece of information you will need is the IP address or hostname of your Raspberry Pi.

Tip

If Wi-Fi is disabled/not connected, you can enable it via the Menu | Preferences | WiFi Configuration link on the desktop

You can obtain this by looking at your local home router, or by running the following commands in the terminal window of Raspbian.

For the IP address, run this command:

- `sudo ip addr show`

Where to look depends on whether you are using a wireless or wired connection. Ethernet can be found usually at eth0 and wireless at wlan0.

If you would prefer to see the hostname you can run this command:

- `sudo hostname`

Make a note of this information, as you will need it to connect to the Raspberry Pi 2 from your second device.

If you are using a Mac or Linux to SSH into the Raspberry Pi you can generate the RSA key via the terminal window using the following steps. If you are using Windows, skip to the relevant section further on in this chapter.

Linux and Mac RSA key generation

Start by opening up your Mac or Linux terminal. From the command line run the following command:

- `ssh-keygen -t rsa -b 4096 -C "username"`

You should replace the username with your own. A message similar to the following will be displayed:

- Generating public/private rsa key pair.

Following this you will see a prompt:

- Enter file in which to save the key (`/Users/username/.ssh/id_rsa`):

You can press enter here and the key will be saved to the path listed in the prompt. Note that it may look slightly different to the preceding example depending on your username and operating system.

Following this, you have the option of adding a password to the RSA private key. These prevent unauthorized users of your second computer from accessing the Raspberry Pi.

- Enter passphrase (empty for no passphrase):
- Enter same passphrase again:

Once you have added a passphrase, the key generation process is complete. You should now see your key's fingerprint:

- Your identification has been saved in `/Users/username/.ssh/id_rsa`.
- Your public key has been saved in `/Users/username/.ssh/id_rsa.pub`.
- The key fingerprint is:

Now we have our key, we need to load it into ssh-agent and then copy it onto the Raspberry Pi 2.

Adding the key to the agent can be done with the following command:

- `ssh-add ~/.ssh/id_rsa`

This now allows the SSH command to use your key when trying to authenticate.

Tip

If your ssh-agent isn't running, you can use the following command to start it:

- eval "\$(ssh-agent -s)"

Before we can SSH into the Raspberry Pi we need to add the public key you created to it.

Thankfully, we can do this in a single command using the Raspberry Pi's default username and password, or if you changed it, that username and password. Run the following command from your terminal. Remember to swap the IP address in the command below with the IP address or hostname you recorded earlier:

- cat ~/.ssh/id_rsa.pub | ssh user@ip'cat >> .ssh/authorized_keys'

With the public key now located on the Raspberry Pi you can attempt to SSH in.

Tip

If the .ssh directory and authorized_keys file do not exist on your Raspberry Pi you can create them under the pi user. To create the directory type mkdir .ssh and to create an empty file in this directory type touch authorized_keys.

This can be achieved using the SSH command along with your username and IP address:

- ssh -A username@ip

You will notice that you can now login to the computer remotely and will be presented with a terminal window prompt.

Windows RSA key generation

In order to connect to the Raspberry Pi from a Windows device you will need three pieces of software, Pageant, PuTTY, and PuTTYgen.

Start by downloading PuTTYgen from the following URL:

<http://the.earth.li/~sgtatham/putty/latest/x86/puttygen.exe>.

Next download PuTTY from this URL:

<http://the.earth.li/~sgtatham/putty/latest/x86/putty.exe>. Like PuTTYgen, this is also an executable you can run from your desktop.

Then finally grab Pageant from this URL:

<http://the.earth.li/~sgtatham/putty/latest/x86/pageant.exe>.

We will start by generating our public and private key pair. Open up the PuTTYgen executable.

Once you have this open, run through the following steps:

Set the key type as SSh-2 RSA.

Click on the Generate button.

You'll now be asked to move your mouse around to generate some random data.

Give your key a passphrase.

Next, use the Save private key button to save the generated private key.

Finally, click the Save public key button.

Our next task is going to be to add the key to the Raspberry Pi 2's authorized_keys file.

Open up the PuTTY executable.

From the Category list on the left, select Session if this is not already open. Now add the following details:

1. In the Host Name field, enter the Raspberry Pi's IP address.
2. Set the Port to 22.
3. Select the SSH radio button.
4. You can now optionally save these details for future connections.
5. Click the Open button.

If this is your first connection, you should now see a popup appear called PuTTY Security Alert. Click the Yes button to move on.

If you entered the connection details successfully, the PuTTY terminal window will now present you with a login prompt for the Raspberry Pi.

You will need to enter in the login name here. By default, this is set as pi; however, you may have changed it if you setup the device via NOOBS.

Following this you will be prompted for the password. This will be raspberry by default, or whichever password you set if you configured Raspbian via NOOBS.

If your login was successful, you should now see the Raspbian command line prompt.

We now want to edit the authorized_keys file. We can use the default text editor installed by Raspbian to edit this file. If the file does not exist, you can create it and the .ssh directory.

It is located under the pi user account:

- .ssh/authorized_keys'

Copy and paste the public key you saved from PuTTYgen into this file.

Save the file and exit it.

You can now logout of the Raspberry Pi.

The final tool we need to test is the Pageant application. This is our windows SSH authentication agent.

Open up the Pageant application. It should be available in the system tray in Windows.

Next, follow these steps:

1. Right click on the Pageant icon.
2. Select Add Key from the menu.
3. A pop-up will display listing any keys you have.
4. Select the Add Key button.
5. From the pop-up window, select the private key you generated with PuTTYgen.
6. Click the Open button.
7. Next, you should be prompted to enter the passphrase for your key.
8. Fill this in and click OK.
9. You should now see it listed in the Pageant Key List window.
10. You can now close the key list.

Whenever you try and access the Raspberry Pi via PuTTY, all you have to enter is the username and host.

Open up PuTTY and connect to the Raspberry Pi again. You should now see you are logged in without a password prompt appearing. This is because you have authenticated your private key against your public key, which was added to the authorized_keys list.

This completes setting up access to the Raspberry Pi remotely. We can now add a static IP address and run some diagnostic tests on our device.

We shall now move onto running some tests on the Raspberry Pi via Raspbian. These can be performed either directly on the Pi via the terminal window, or over the SSH connection you just created.

Assign a static IP to your Raspberry Pi 2

Assigning a static IP address to your Raspberry Pi means that when you switch it on or off a new IP will not be assigned. Instead, it will always contain the same IP address, meaning you do not have to hunt down the value assigned to it by the DHCP server each time you reboot.

To start with, check the IP address range on your router and find a free IP address. You will also need the subnet mask and the default gateway.

Next, we need to assign the free address to the Raspberry Pi. Editing the interfaces file can do this:

- sudo nano /etc/network/interfaces

In the open file you will need to locate the line that specifies eth0 or wlan0 depending on whether you are wired or wireless. For example:

- iface eth0 inet dhcp

Change the value dhcp to static:

- iface eth0 inet static

Once this is done we need to add three lines directly below it specifying the IP address we wish to assign, netmask, and gateway. You should have these values from checking your router earlier.

Paste these in below the interface. **An example is shown here:**

- address 192.168.1.132
- netmask 255.255.255.0
- gateway 192.168.1.1

Save the file using Ctrl + X and press Y to save.

We do not need to reboot the Raspberry Pi to apply these changes, but can stop and **start the network interface using the following commands:**

- sudo /etc/init.d/networking stop
- sudo /etc/init.d/networking start

If you now run the command from earlier to check the IP address, you should see it is the new value you assigned:

- sudo ip addr show

In the preceding example we used the nano text editing tool. We shall now look at some other options for editing files.

OR

We can easily use **VNC server** its free in linux software after we give RPI static ip address and connect it to our switch or access point this is in RPI side in PC side we download **VNC Viewer** and connect our PC to the same switch with the same ip address mask and other host number to the network this make us could fully control on RPI with its GUI.

This is raspberry pi 2 model B in 2014 and there is many models before and after that so



Figure 3-2. Raspberry Pi models

Let's take a look to some of these boards and show the difference between them:

Table 3-2. Raspberry Pi models Comparison

Model	RPi 2 B	RPi 3 B	RPi 3 B+	RPi 4 B
SOC Type	Broadcom BCM2836	Broadcom BCM2837	Broadcom BCM2837B0	Broadcom BCM2711
CPU Clock	4 × Arm Cortex-A7, 900 MHz	4 × Arm Cortex-A53, 1.2 GHz	4 × Arm Cortex-A53, 1.4 GHz	4 × Arm Cortex-A72, 1.5 GHz
RAM	1 GB	1 GB	1 GB	1 GB/2 GB/4 GB
GPU	Broadcom VideoCore IV	Broadcom VideoCore IV	Broadcom VideoCore IV	Broadcom VideoCore VI
USB Ports	4	4	4	4 (2 × USB 3.0 + 2 × USB 2.0)
Ethernet	100 Mbit/s base Ethernet	100 Mbit/s base Ethernet	Gigabit Ethernet (max. 300 Mbps)	Gigabit Ethernet (no limit)
Power over Ethernet	No	No	Yes (requires separate PoE HAT)	Yes (requires separate PoE HAT)
WiFi	No	WiFi 802.11n	WiFi 802.11ac Dual Band	WiFi 802.11ac Dual Band
Bluetooth	No	4.1	4.2 BLE	5.0 BLE
Video Output	HDMI/3.5 mm Comp./DSI	HDMI/3.5 mm Comp./DSI	HDMI/3.5 mm Comp./DSI	micro-HDMI/3.5 mm Comp./DSI
Audio Output	I ² S/HDMI/3.5 mm Composite			
Camera Input	15 Pin CSI	15 Pin CSI	15 Pin CSI	15 Pin CSI
GPIO Pins	40	40	40	40
Memory	MicroSD	MicroSD	MicroSD	MicroSD

Our project raspberry pi model

Description

3.3 Raspberry Pi 4 B

version with 4GB memory and layout of power supply components updated.

Price: US\$75.00.

SOC Type: Broadcom BCM2711.

Core Type: Cortex-A72 (ARM v8) 64-bit.

No. Of Cores: 4.

GPU: Video Core VI.

CPU Clock: 1.5 GHz.

RAM: 4 GB LPDDR4.

USB: 2x USB3.0 + 2x USB2.0 + USB-C OTG.

Ethernet: Gigabit.

HDMI port: 2x micro HDMI.

Analog Video Out: shared with audio jack.

Analog Audio Out: 3.5mm jack.

Analog Audio In: via sound card.

GPIO: 40-pins.

LCD Panel: exist and we will talk about it in detail in this chapter.

SD/MMC: microSD.

Camera: exist and we will talk about it in detail in this chapter.

Wireless Connectivity (On-Board)

Wi-Fi: 2.4GHz and 5GHz 802.11 b/g/n/ac.

Bluetooth®: 5.0.

Dimensions

Height: 2.22 in (56.5 mm).

Width: 3.37 in (85.6 mm).

Depth: 0.43307 in (11 mm).

Weight: 1.62 oz (46 g).

Power

Power ratings: 1.25 A @5V.

Power sources: USB-C.

Front view:



Figure 3-3. Raspberry Pi 4 Front view

3.3.1 LCD:

Models of LCDs:

- 15.6inch HDMI LCD (H) (with case)
- 15.6inch HDMI LCD (H)
- 15.6inch FHD Monitor
- 13.3inch HDMI LCD (H) (with case)
V2
- 13.3inch HDMI LCD (H) (with case)
- 13.3inch HDMI LCD (H)
- 13.3inch Magic Mirror
- 13.3inch PiPad
- 12.5inch FHD Monitor

- 11.6inch HDMI LCD (H) (with case)
- 11.6inch HDMI LCD (H)
- 11.9inch HDMI LCD
- 10.5inch HDMI AMOLED

- 10.1inch HDMI LCD (G) (with case)
- 10.1inch HDMI LCD (H) (with case)
- 10.1HP-CAPQLED
- 9inch 2560x1600 Monitor
- 9HP-CAPQLED
- 8.8inch Side Monitor
- 8inch DSI LCD
- 7.9inch HDMI LCD
- 7inch FHD Monitor
- 7inch HDMI LCD (H) (with case)
- 7inch HDMI LCD (C)
- 7inch HDMI LCD (B)
- 7inch HDMI LCD
- 7inch DPI LCD
- 7HP-CAPQLED
- 7inch DSI LCD

- 5inch HDMI LCD (B)
 - 5inch HDMI LCD
 - 5inch HDMI AMOLED
 - 5inch DPI LCD
 - 5inch DSI LCD
 - 5.5inch HDMI AMOLED
 - 4.3inch HDMI LCD (B)
 - 4.3inch DSI LCD
 - 4inch DPI LCD (B)
 - 4inch DPI LCD (C)
 - 4inch HDMI LCD (H)
 - 4inch HDMI LCD
 - 4inch RPi LCD (C)
 - 4inch RPi LCD (A)
 - 3.5inch DPI LCD
 - 3.5inch HDMI LCD
 - 3.5inch RPi LCD (C)
- 2.23inch OLED HAT
 - 1.3inch OLED HAT
 - 1.3inch LCD HAT
 - 1.44inch LCD HAT
 - 1.8inch LCD Module
 - EINK-DISP-103
 - LCD1602 RGB Module
 - Pico-Clock-Green
 - Pico-RGB-Matrix-P3-64x32
 - RGB-Matrix-P3-64x32

We choose our type as 3.5 inch Touch Screen TFT LCD Designed for Raspberry Pi (this type of LCD connected using PIN headers NOT HDMI).



Figure 3-4. LCD model



Figure 3-5. LCD

Features

1. 480×320 hardware resolution
2. Resistive touch control
3. It is compatible and can be directly inserted in any version of Raspberry Pi.
4. IPS screen with large viewing angle and outstanding display effect.

5. Drivers provided (works with your own Raspbian/Ubuntu/Kali and Retropie system directly).
6. Support FBCP software driver, can set the software resolution and dual screen display.
Gold sinking process, fine carving.

Getting Started

Hardware Connection

There are 40 pins on Raspberry Pi but only 26 pins on the LCD, so you should pay attention to connecting the pins to your Pi accordingly.

3.3.2 Install the touch driver

The RPi LCD can be driven in two ways: Method 1. install driver to your Raspbian OS. Method 2. use the Ready-to-use image file of which LCD driver was pre-installed.

Method 1.

Driver installation

Please download the latest version of the image on the Raspberry Pi official website.

(<https://www.raspberrypi.org/downloads/>) (Raspbian/Ubuntu Mate/Kali or Retropie)

- 1) Download the compressed image file to the PC, and unzip it to get the .img file.
- 2) Connect the TF card to the PC, open the Win32DiskImager software, select the system image downloaded in step 1 and click 'Write' to write the system image. (How to write an image to a micro SD card for your Pi? See RPi Image Installation Guides for more details)
- 3) Connect the TF card to the Raspberry Pi, start the Raspberry Pi. The LCD will display after booting up, and then log in to the Raspberry Pi terminal.(You may need to connect a keyboard and HDMI LCD to Pi for driver installing, or log in remotely with SSH)
- 4) Then open the terminal of Raspberry Pi to install the touch driver.

```
git clone https://github.com/waveshare/LCD-show.git
cd LCD-show/
```

Note: The Raspberry Pi must be connected to the network, or else the touch driver won't be successfully installed.

```
#if old version, use this command:
```

```
chmod +x LCD35B-show
./LCD35B-show
```

```
#if new version, use this command:
```

```
chmod +x LCD35B-show-V2
./LCD35B-show-V2
```

The touch function will work after restart. For ease of use, you can set the screen orientation,

see:

#Screen orientation settings.

Notes:

1. Executing apt-get upgrade will cause the LCD to fail to work properly. In this case, you need to edit the config.txt file in the SD card and delete this sentence: dtoverlay=ads7846.
2. Using with Raspbian-lite, the command of the touch driver installation should be:

```
#if old version, use this command:
./LCD35B-show lite
```

```
#if new version, use this command:
./LCD35B-show-V2 lite
```

Method 2. Using Ready-to-use image

The image file with the pre-installed driver is located in the IMAGE directory of the CD, or you can download it from #Image. Extract the .7z file and you will get an .img file. Write the image to your micro SD card (How to write an image to a micro SD card for your Pi? See RPi Image Installation Guides for more details). Then insert the card to your Pi, power up, and enjoy it.

Toggle between LCD and HDMI display

Once this LCD is enabled, meanwhile the default settings for HDMI are changed. If you want to use another HDMI monitor, please run the following command:

```
cd LCD-show/  
./LCD-HDMI
```

This toggles the mode to LCD display:

```
#if old version, use this command:  
./LCD35B-show  
  
#if new version, use this command:  
./LCD35B-show-V2
```

3.3.3 Screen orientation settings

After touch driver is installed, the screen orientation can be set by these commands:

0 degree rotation

```
cd LCD-show  
#if old version, use this command:  
./LCD35B-show 0  
#if new version, use this command:  
./LCD35B-show-V2 0
```

90 degree rotation

```
cd LCD-show/  
  #if old version, use this command:  
  ./LCD35B-show 90  
  #if new version, use this command:  
  ./LCD35B-show-V2 90
```

180 degree rotation

```
cd LCD-show/  
  #if old version, use this command:  
  ./LCD35B-show 180  
  #if new version, use this command:  
  ./LCD35B-show-V2 180
```

270 degree rotation

```
cd LCD-show/  
  #if old version, use this command:  
  ./LCD35B-show 270  
  #if new version, use this command:  
  ./LCD35B-show-V2 270
```

3.3.4 Touch screen calibration

This LCD can be calibrated through the xinput-calibrator program. **Note: The Raspberry Pi must be connected to the network, or else the program won't be successfully installed.**

Run the following command to install:

```
sudo apt-get install xinput-calibrator
```

Click the "Menu" button on the taskbar, choose "Preference" -> "Calibrate Touchscreen".

Finish the touch calibration following the prompts. Maybe rebooting is required to make calibration active.

You can create a 99-calibration.conf file to save the touch parameters (not necessary if file exists).

```
sudo mkdir /etc/X11/xorg.conf.d
sudo nano /etc/X11/xorg.conf.d/99-calibration.conf
```

Save the touch parameters (may differ depending on LCD) to 99-calibration.conf, as shown in the picture:

```
Section "InputClass"
    Identifier      "calibration"
    MatchProduct   "ADS7846 Touchscreen"
    Option        "Calibration"  "208 3905 288 3910"
    Option        "SwapAxes"     "0"
EndSection
```

Press the keys Ctrl+X, and select option Y to save the modification. The modification will be valid after rebooting the system. Enter the following command for system reboot:

```
sudo reboot
```

3.3.5 Install Virtual Keyboard

1. Install matchbox-keyboard

```
sudo apt-get install update
sudo apt-get install matchbox-keyboard
sudo nano /usr/bin/toggle-matchbox-keyboard.sh
```

2. Copy the statements below to toggle-matchbox-keyboard.sh and save.

```
#!/bin/bash
#This script toggle the virtual keyboard
PID=`pidof matchbox-keyboard`
if [ ! -e $PID ]; then
    killall matchbox-keyboard
else
    matchbox-keyboard -s 50 extended&
fi
```

3. Execute the commands:

```
sudo chmod +x /usr/bin/toggle-matchbox-keyboard.sh
sudo mkdir /usr/local/share/applications
sudo nano /usr/local/share/applications/toggle-matchbox-keyboard.desktop
```

4. Copy the statements to toggle-matchbox-keyboard.desktop and save.

```
[Desktop Entry]
Name=Toggle Matchbox Keyboard
Comment=Toggle Matchbox Keyboard
Exec=toggle-matchbox-keyboard.sh
Type=Application
Icon=matchbox-keyboard.png
Categories=Panel;Utility;MB
X-MB-INPUT-MECHANSIM=True
```

5. Execute commands as below. Note that you need to use "Pi " user permission instead of root to execute this command

```
nano ~/.config/lxpanel/LXDE-pi/panels/panel
```

6. Find the statement which is similar to below: (It maybe different in different version)

```
Plugin {
    type = launchbar
    Config {
        Button {
            id=lxde-screenlock.desktop
        }
        Button {
            id=lxde-logout.desktop
        }
    }
}
```

7. Append these statements to add an button option:

```
Button {
    id=/usr/local/share/applications/toggle-matchbox-keyboard.desktop
}
```

```
Plugin {
    type=space
    Config {
        Size=8
    }
}
Plugin {
    type=launchbar
    Config {
        Button {
            id=/usr/local/share/applications/toggle-matchbox-keyboard.desktop
        }
        Button {
            id=/usr/share/applications/lxde-x-www-browser.desktop
        }
        Button {
            id=/usr/share/raspi-ui-overrides/applications/pcmanfm.desktop
        }
        Button {
            id=/usr/share/raspi-ui-overrides/applications/lxterminal.desktop
        }
        Button {
            id=/usr/share/applications/wolfram-mathematica.desktop
        }
        Button {
            id=/usr/share/applications/wolfram-language.desktop
        }
    }
}
Plugin {
    type=space
    Config {
        Size=8
    }
}
```

8. reboot your Raspberry Pi. If the virtual keyboard is installed correctly, you can find that there is a keyboard icon on the left of the bar

`sudo reboot`

NOTE: PWM adjust brightness (for 2.8inch RPi LCD (A) only)

Update wiring Pi libraries if you use Pi 4

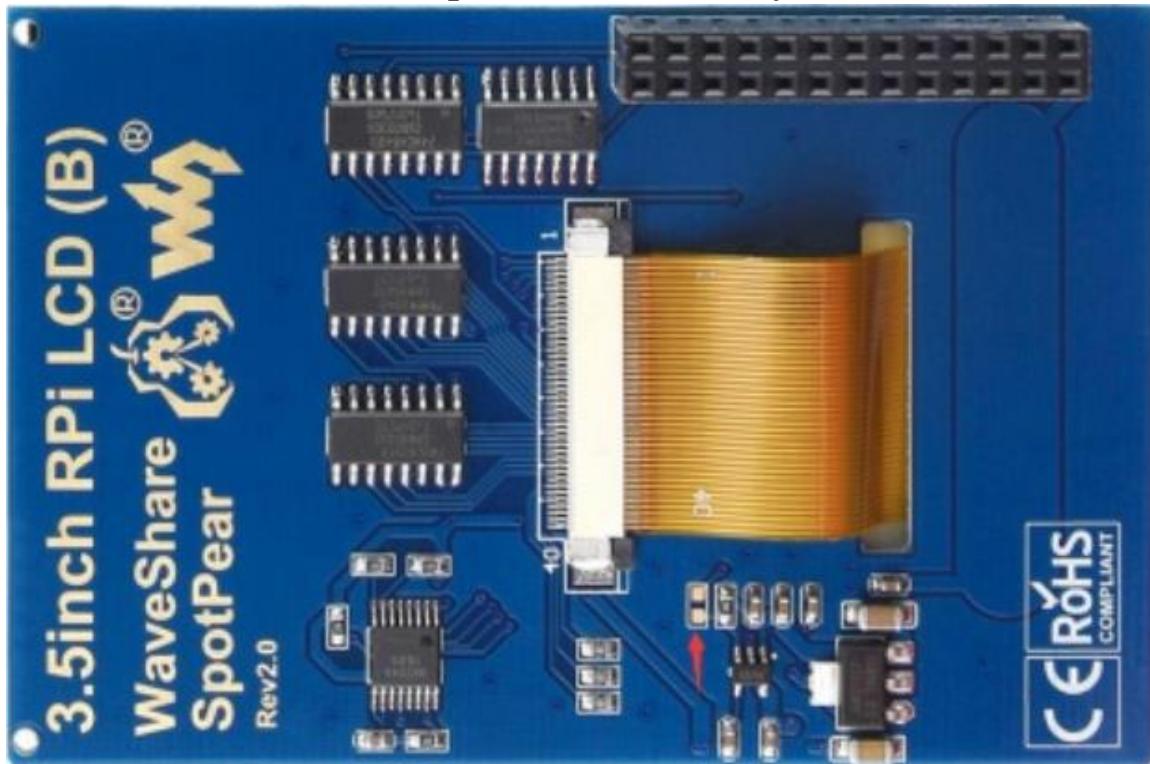
```
wget https://project-downloads.drogon.net/wiringpi-latest.deb  
sudo dpkg -i wiringpi-latest.deb  
gpio -v
```

Adjust the brightness with the following commands:

```
gpio -g mode 18 pwm      #Set the pin as PWM  
gpio pwmc 100  
gpio -g pwm 18 0          #Set the brightness to dimmest  
gpio -g pwm 18 1023       #Set the brightness to the brightest  
gpio -g mode 18 out        #Clean the pin
```

Use GPIO to control backlight brightness

Solder a 0R resistor on this pad or solder directly to connect.



Execute the following commands in the Raspberry Pi terminal:

```
gpio -g pwm 18 1024
gpio -g mode 18 pwm
gpio pwmc 1000
```

Control brightness:

```
gpio -g pwm 18 X
```

X ranges from 0 to 1024, 0 is the darkest, 1024 is the brightest

Interface:

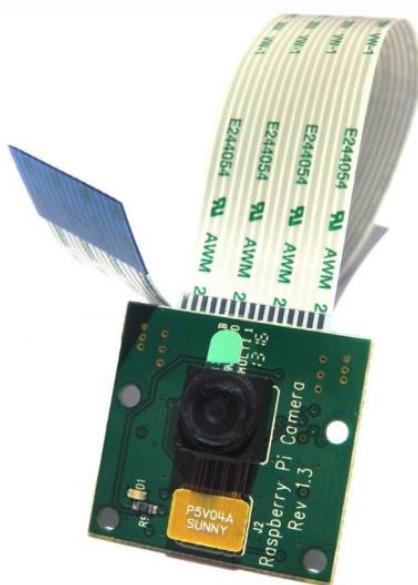
Table 3-3. Raspberry Pi Pins Description

PIN NO.	SYMBOL	DESCRIPTION
1, 17	3.3V	Power positive (3.3V power input)
2, 4	5V	Power positive (5V power input)
3, 5, 7, 8, 10, 12, 13, 15, 16	NC	NC
6, 9, 14, 20, 25	GND	Ground
11	TP IRQ	Touch Panel interrupt, low level while the Touch Panel detects touching
18	LCD_RS	Command/data register selection
19	LCD_SI / TP_SI	LCD display/SPI data input of Touch Panel
21	TP_SO	SPI data output of Touch Panel
22	RST	Reset
23	LCD_SCK / TP_SCK	LCD display/SPI clock of Touch Panel
24	LCD_CS	LCD chip selection, low active
26	TP_CS	Touch Panel chip selection, low active

3.3.6 Raspberry pi Camera:

Raspberry pi camera rev 1.3 5M PIXEL is our camera module in this project.

Description:



The Raspberry Pi Camera Board plugs directly into the CSI connector on the Raspberry Pi. It's able to deliver a crystal clear 5MP resolution image, or 1080p HD video recording at 30fps! Latest Version 1.3! Custom designed and manufactured by the Raspberry Pi Foundation in the UK, the Raspberry Pi Camera Board features a 5MP (2592?1944 pixels) Omnivision 5647 sensor in a fixed focus module. The module attaches to Raspberry Pi, by way of a 15 Pin Ribbon Cable, to the dedicated 15-pin MIPI Camera Serial Interface (CSI), which was designed especially for interfacing to cameras. The CSI bus is capable of extremely high data rates, and it exclusively carries pixel data to the BCM2835 processor. The board itself is tiny, at around 25mm x 20mm x 9mm, and weighs just over 3g, making it perfect for mobile or other applications where size and weight are important. The sensor itself has a native resolution of 5 megapixel, and has a fixed focus lens onboard. In terms of still images, the camera is capable of 2592 x 1944 pixel static images, and also supports 1080p @ 30fps, 720p @ 60fps and 640x480p

60/90 video recording. The camera is supported in the latest version of Raspbian, the Raspberry Pi's preferred operating system. The Raspberry Pi Camera Board Features:

1. Fully Compatible with Both the Model A and Model B Raspberry Pi
2. 5MP Omnivision 5647 Camera Module
3. **Still Picture Resolution:** 2592 x 1944
4. **Video:** Supports 1080p @ 30fps, 720p @ 60fps and 640x480p 60/90 Recording
5. 15-pin MIPI Camera Serial Interface - Plugs Directly into the Raspberry Pi Board
6. **Size:** 20 x 25 x 9mm
7. Weight 3g Fully Compatible with many Raspberry Pi cases

How to work with a camera How to work with a camera

1. Select "Enable Camera" -> "<YES>"

```
sudo raspi-config
```

2. Copy the Camera driver to the OS of Pi then:

```
unzip camera.zip
cd camera
sudo chmod 777 Camera
sudo cp update\ camera/95-stmpe.rules /etc/udev/rules.d/
```

3. Create a file called wheezy.list.

```
sudo nano /etc/apt/sources.list.d/wheezy.list
```

Append:

```
deb http://archive.raspbian.org/raspbian wheezy main
```

Exit with save.

4. Create a file called 10defaultRelease.

```
sudo nano /etc/apt/apt.conf.d/10defaultRelease
```

Append:

```
APT::Default-release \"stable\";
```

Exit with save.

5. Create a file called libsdl.

```
sudo nano /etc/apt/preferences.d/libsdl1
```

Append:

```
Package: libsdl1.2debian
Pin: release n=jessie
Pin-Priority: -10
Package: libsdl1.2debian
Pin: release n=wheezy
Pin-Priority: 900
```

6. Last, execute the commands:

```
sudo apt-get update
sudo apt-get -y --force-yes install libsdl1.2debian/wheezy
sudo apt-get install evtest tslib libts-bin xinput
sudo apt-get install python-pip
sudo apt-get install python2.7-dev
sudo pip install picamera==1.10
sudo reboot
sudo TSLIB_FBDEVICE=/dev/fb1 TSLIB_TSDEVICE=/dev/input/touchscreen ts_calibrate
```

Camera will be enabled by the steps above.

We also can use **normal computer mouse as peripheral instead of the use of touch option in the touch screen.**



32GB SanDisk Micro SDHC Memory



3.4 Implementation:

We have different modules to be implemented in our project **Exam module**, **Face recognition module**, **Question module**, **Vote module** and **Quiz module**.

Ex:

When we receive message from server in main to start the **exam** so first we use **facial recognition** to detect the student who intend to have the exam and start the Exam from Exam module and then send the score, all of this modules are separated and main that will be to handle all of this.

3.5 Main Module:

3.5.1 Websocket protocol

WebSocket is a computer communications protocol, providing full-duplex communication channels over a single TCP connection, between student and server.

This connection must be in both direction (bidirectional communication), bidirectional communications means

- Transmitting data in both directions, client to server and server to client but not at the same time.

- This provides the ability to enable real-time connection between client and server with TCP connection.

At first they used HTTP as transport layer but this method was not efficient because client didn't take any response from server if it didn't make a request first to have a response from server because server will not send response from itself this method makes client to make poll request frequently this called short poll

- **HTTP short poll example**

1. **Client** at **5:00:00 PM** : Is there any notification?
2. **Server** at **5:00:00 PM** : Not yet...
3. **Client** at **5:00:00 PM** : Is there any notification?
4. **Server** at **5:00:00 PM** : Not yet...
5. **Client** at **5:00:01 PM** : Is there any notification?
6. **Server** at **5:00:01 PM** : Yes, I will send a response
7. **Client** at **5:00:02 PM** : Thanks, I am sending new Poll request

- **HTTP short poll problems**

1. Not efficient.
2. High latency.
3. More resources are needed.

Long polling life cycle another method used to solve problems we found at short polling, in this method server receives request and still opened this called hold open and if there is any data at server it will transmit it to client but also this method have a problems.

- **HTTP long poll example**

1. **Client** at **5:00:00 PM** : Is there any notification?
2. **Server** at **5:00:05 PM** : Yes, I will send a response
3. **Client** at **5:00:05 PM** : Thanks, I am sending new Poll request, Is there any notification?
4. **Server** at **5:00:50 PM** : Yes, I will send a response

- **HTTP long poll problems**
 1. Transferred data may be longer than actual data.
 2. Packet loss.
 3. Frequent opening and closing TCP connection this may cause DOS attack.
 4. Latency reduced, but still not efficient due to issues that were mentioned before.
- **HTTP as transport problems (long and short polling) issues conclusion**
 1. Multiple TCP connection for each client.
 2. Transferred data is bigger than actual data.
 3. Mapping the outgoing to incoming connection.
 4. DOS attack is possible.
 5. HTTP is not meant to be used for bidirectional communication.

To fix these issues we use bidirectional communications with single TCP connection **by using websocket protocol**.

Note that from now we are talking about websocket protocol NO LONG OR SHORT POLLING.

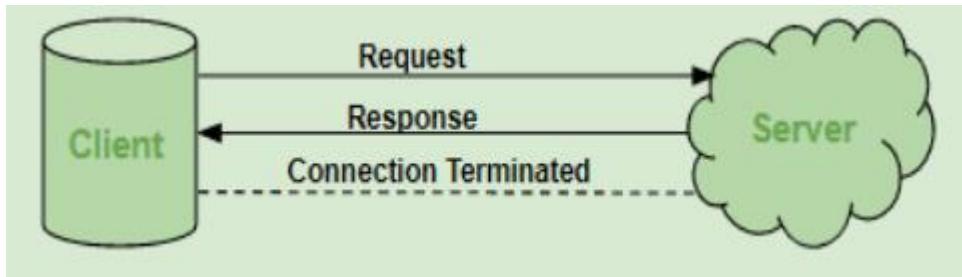


Figure 3-6: HTTP communication

- **Websocket**

A WebSocket server is nothing more than an application listening on any port of a TCP server that follows a specific protocol. The task of creating a custom server tends to scare people; however, it can be straightforward to implement a simple WebSocket server on your platform of choice.

A WebSocket server can be written in any server-side programming language such as C(++), Python, PHP, or server-side JavaScript.

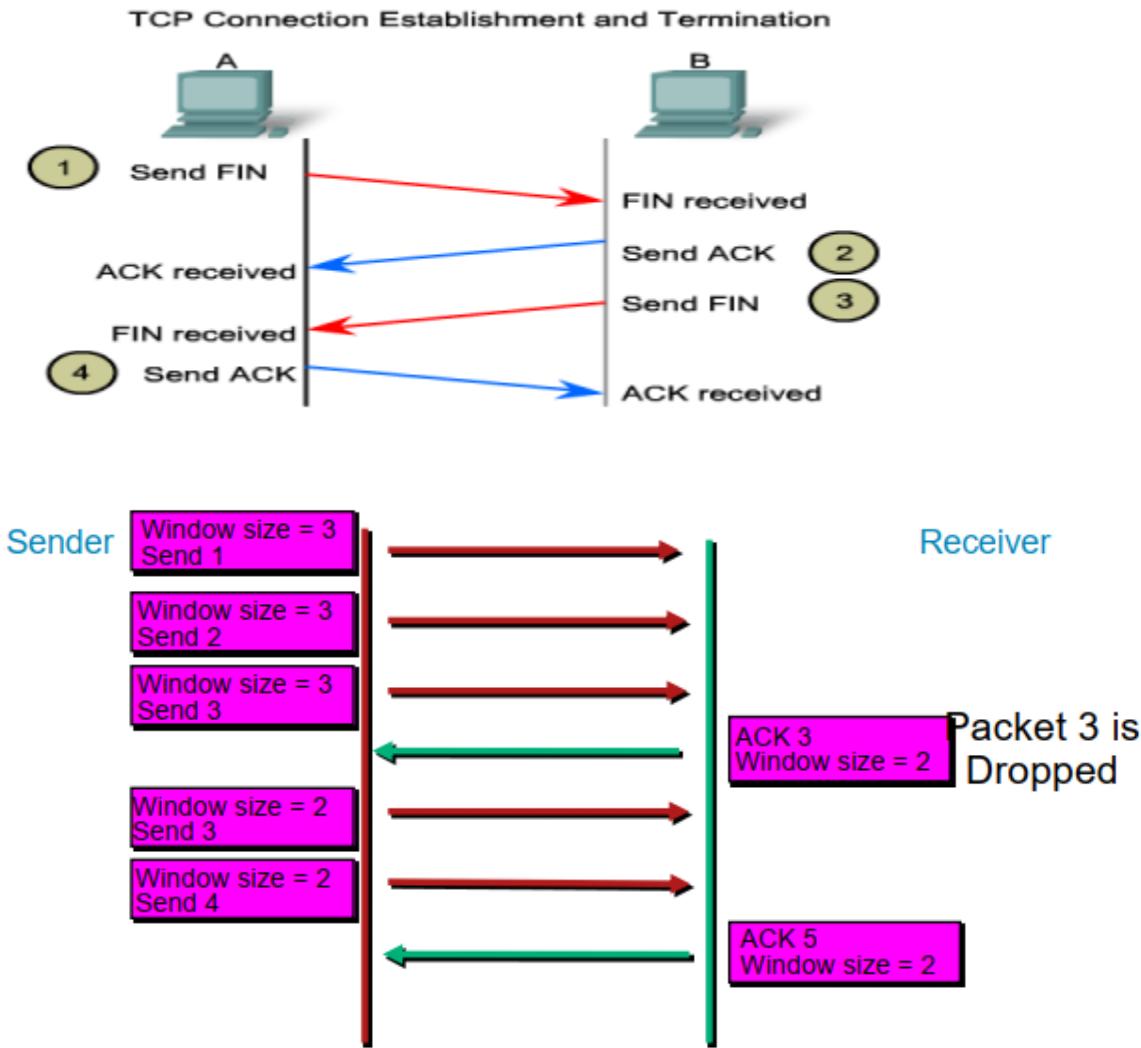
Depending on language support, knowledge of TCP sockets may be required.

- **Websocket features**

1. Based on hand shacking.
2. Runs on ports like 80 and 443 for secure connections.
3. Using single TCP connections.
4. Designed to be frame-based.
5. Supports Unicode and binary frames.
6. Websocket is a layer on top of TCP.
7. Multiple services, host names one ip address.
8. Independent TCP-based protocol uses HTTP only at hand shacking.

The Transmission Control Protocol (TCP)

TCP is [connection-oriented](#), and a connection between client and server is established before data can be sent. The server must be listening (passive open) for connection requests from clients before a connection is established.



3.6 WebSocket implementation in our project:

Libraries included in this part of project

```
4 import json
5 import asyncio
6 import websockets
```

In this part of code we used function “progress” used for the continuous connection with server as the client is waiting until get message from server, after that we pass this message to “check” where it will be analyzed to see

who send this message and the destination and there valid action to run function in the end device like (open_cam_for_admin, train) for administrator,

(open_cam_for_prof, mute_all, vote,start_session, end_session, cheat) for the professor,

Professor connects to server:

```
{  
    "action" : "connect" ,  
    "to"     : "server" ,  
    "from"   : "professor" ,  
    "device_id": "99999" ,  
    "execute" : {  
        "token": "12341351325123413513251234135321" ,  
        "name": "Ahmed Ragab"  
    }  
}
```

Server response:

```
{  
    "action": "response",  
    "device_id": 5252,  
    "from": "server",  
    "to": "professor",  
    "execute": {  
        "status": "DUPLICATE_CONNECTION"  
    }  
}
```

```
{  
    "action": "response",  
    "from": "server",  
    "to": "professor",  
    "device_id": "99999",  
    "execute": {  
        "status": "OK"  
    }  
}
```

Vote process:

```
{  
    "action": "vote",  
    "to": "student",  
    "from": "professor",  
    "student_name": "hosam",  
    "device_id": "device_ID",  
    "execute": {  
        "question": "who are?",  
        "first": "hosam",  
        "second": "ali",  
        "third": "alla",  
        "fourth": "hady"  
    }  
}  
  
{  
    "action": "done voting",  
    "to": "professor",  
    "from": "student",  
    "student_name": "xxxxxx",  
    "device_id": "xxxxxx",  
    "execute": {  
        "student_id": "studend_id",  
        "device_id": "device_ID",  
        "student_name": "xxxxxx",  
        "choice": "xxxxxx"  
    }  
}  
  
{  
    "action": "failed",  
    "to": "professor",  
    "from": "student",  
    "student_name": "xxxxxx",  
    "device_id": "xxxxxx",  
    "execute": {  
        "student_id": "xxxxxx",  
        "device_id": "xxxxxx",  
        "student_name": "xxxxxx",  
        "cause": "not in data base"  
    }  
}
```

Mute process from the professor:

```
{  
    "action": "mute_all",  
    "to": "student",  
    "from": "professor",  
    "student_name": "xxxxxxxx",  
    "device_id": "xxxxxxxx",  
    "execute": "none"  
}  
  
{  
    "action": "done muting",  
    "to": "professor",  
    "from": "student",  
    "student_name": "sadfdslf",  
    "device_id": "xxxxxx",  
    "execute": {  
        "device_id": "xxxxxx",  
        "student_name": "sadfdslf"  
    }  
}
```

Mute process student himself:

```
{  
    "action": "mute",  
    "to": "student",  
    "from": "student",  
    "student_name": "hosam",  
    "device_id": "xxxxxxxxxx",  
    "execute": "none"  
}  
  
{  
    "action": "done muting",  
    "to": "student",  
    "from": "student",  
    "student_name": "xxxxxxxx",  
    "device_id": "xxxxxx",  
    "execute": {  
        "student_id": "xxxxxx",  
        "device_id": "xxxx",  
        "student_name": "xxxxxxxx"  
    }  
}  
  
{  
    "action": "failed",  
    "to": "student",  
    "from": "student",  
    "student_name": "abdo",  
    "device_id": "xxxxxx",  
    "execute": {  
        "student_id": "xxxxxx",  
        "device_id": "xxxxxxxx",  
        "student_name": "xxxxxxxx",  
        "cause": "not in data base"  
    }  
}
```

Cheat process:

```
{  
    "action": "cheat",  
    "to": "student",  
    "from": "professor",  
    "student_name": "hosam",  
    "device_id": "xxxxxxxx",  
    "execute": "none"  
}  
  
{  
    "action": "done cheating",  
    "to": "professor",  
    "from": "student",  
    "student_name": "xxxxxxxx",  
    "device_id": "xxxxxx",  
    "execute": {  
        "student_id": "xxxxxxxx",  
        "device_id": "xxxxxxxxxx",  
        "student_name": "xxxxxxxx",  
        "cause": "not in data base"  
    }  
}
```

Start session process:

```
[ {  
    "action": "failed",  
    "to": "professor",  
    "from": "student",  
    "student_name": "xxxx",  
    "device_id": "xxxxx",  
    "execute": {  
        "student_id": "xxxxx",  
        "device_id": "xxxxxxxxx",  
        "student_name": "xxxxxxxx",  
        "cause": "not in data base"  
    }  
}, {  
    "action": "done starting sission",  
    "to": "professor",  
    "from": "student",  
    "student_name": "xxxxxxxxx",  
    "device_id": "xxxxx",  
    "execute": {  
        "student_id": "xxxx",  
        "device_id": "xxxxx",  
        "student_name": "xxxxxx"  
    }  
}]
```

End session Process:

```
{  
    "action": "end_session",  
    "to": "student",  
    "from": "professor",  
    "student_name": "all_student",  
    "device_id": "xxxxxx",  
    "execute": "none"  
}
```

Start exam process:

```
{  
    "action": "start_exam",  
    "to": "student",  
    "from": "professor",  
    "student_name": "hosam",  
    "device_id": device_ID,  
    "execute": [  
        { 'question': 'what is your name?', 'correct_answer': 'Tantawy', 'incorrect_answers': ['Hossam', 'Gad', 'Saber'  
            ]  
        },  
        { 'question': 'How old are you?', 'correct_answer': '22', 'incorrect_answers': ['20', '21', '19'  
            ]  
        },  
        { 'question': 'where are you from?', 'correct_answer': 'benha', 'incorrect_answers': ['giza', 'aswan', 'cairo' →  
            ]  
        },  
        { 'question': 'what is your job?', 'correct_answer': 'still student', 'incorrect_answers': ['engineer', 'doctor', 'teacher'  
            ]  
        },  
        { 'question': 'predicted graduated year?', 'correct_answer': '2022', 'incorrect_answers': ['2021', '2023', '2024'  
            ]  
        },  
        { 'question': 'your best friend?', 'correct_answer': 'Tantawy', 'incorrect_answers': ['sayed', 'saber', 'goda'  
            ]  
        }  
    ]  
}
```

```
{  
    "action": "done exam",  
    "to": "professor",  
    "from": "student",  
    "student_name": "xxxxxxxx",  
    "device_id": "xxxxxxxx",  
    "execute": {  
        "student_id": "xxxxxx",  
        "device_id": "xxxxxxxx",  
        "student_name": "xxxxxxxx",  
        "cause": "not in data base"  
    }  
}
```

```
{  
    "action": "failed",  
    "to": "professor",  
    "from": "student",  
    "student_name": "xxxxxxxx",  
    "device_id": "xxxxxxxx",  
    "execute": {  
        "student_id": "xxxxxx",  
        "device_id": "xxxxxxxxxxxx",  
        "student_name": "xxxxxxxx",  
        "correct": "x",  
        "wrong": "xx",  
        "grade": "xx",  
        "array": "xx"  
    }  
}
```

Open cam for professor:

```
{  
    "action": "open_cam_for_prof",  
    "to": "student",  
    "from": "professor",  
    "student_name": "hossam",  
    "device_id": "device_ID",  
    "execute": "none"  
}
```

```
{  
    "action": "failed",  
    "to": "professor",  
    "from": "student",  
    "student_name": "xxxxxx",  
    "device_id": "xxxxxx",  
    "execute": {  
        "student_id": "xxxxxx",  
        "device_id": "xxxxxx",  
        "student_name": "xxxxxx",  
        "cause": "not in data base"  
    }  
}
```

```
{  
    "action": "failed",  
    "to": "professor",  
    "from": "student",  
    "student_name": "xxxxxxxxx",  
    "device_id": "xxxxxxxxx",  
    "execute": {  
        "student_id": "xxxxxxxx",  
        "device_id": "xxxxxxxx",  
        "student_name": "xxxxxxxxxxxx",  
        "cause": "another student"  
    }  
}
```

```
{  
    "action": "done recognizing",  
    "to": "professor",  
    "from": "student",  
    "student_name": "xxxxxxxx",  
    "device_id": "xxxxxxxx",  
    "execute": {  
        "student_id": "xxxxxxxx",  
        "device_id": "xxxxxxxx",  
        "student_name": "xxxxxxxx"  
    }  
}
```

Open cam for admin

```
{  
    "action": "open_cam_for_admin",  
    "to": "student",  
    "from": "administrator",  
    "student_name": "hossam",  
    "device_id": "xxxxxxxx",  
    "execute": {  
        "student_name": "hossam",  
        "student_ID": 43,  
        "n.o pictures": 15  
    }  
}
```

```
{  
    "action": "done registering",  
    "to": "administrator",  
    "from": "student",  
    "student_name": "xxxxxxxx",  
    "device_id": "xxxxxxxx",  
    "execute": {  
        "student_id": "xxxxxxxx",  
        "device_id": "xxxxxxxx",  
        "student_name": "xxxxxxxxxxxx"  
    }  
}
```

Train process:

```
{  
    "action": "train",  
    "to": "student",  
    "from": "administrator",  
    "student_name": "all_student",  
    "device_id": "xxxxxxxx",  
    "execute": "none"  
}
```

```
{  
    "action": "done trainning",  
    "to": "administrator",  
    "from": "none",  
    "student_name": "none",  
    "device_id": "none",  
    "execute": "none"  
}
```

3.7 Exam:

When the professor wants to create an exam for the students,

The professor sends the questions, three wrong choices for each question and the correct answer for each question.

Using the python GUI, we represent this information for student.

Firstly, we split the message of the professor into 3 variables for each question:

```
for question in question_data:  
    question_text = question["question"]  
    correct_answer=question["correct_answer"]  
    incorrect_answers=question["incorrect_answers"]
```

then we create a variable called choices to contain both correct and incorrect answers and shuffle them

```
for ans in incorrect_answers:  
    choices.append(ans)  
choices.append(correct_answer)  
shuffle(choices)
```

then we create a new form of each question containing: (question_text, correct_answer, choices)

and we append all questions in a question bank:

```
new_question = Question(question_text, correct_answer, choices)  
question_bank.append(new_question)
```

Now in the python GUI we want to represent some information:

1. The title of the exam
2. The question

3. The four choices
4. Four radio buttons in front of each choice
5. A next button to go to the next question
6. A previous button to go to the previous question

Before we talking about the quiz GUI there is some functions that we would use in actions of GUI. We write them in "quiz_brain.py":

1. has_more_question:

To check if the quiz has more questions, if there is the next button go to the next question. If not, the next button terminate the exam. If the current question number is less than number of questions it returns 1 else it returns 0.

```
def has_more_questions(self):  
    return self.question_no < len(self.questions)
```

Exam model

Q.6: your best friend?

- goda
- saber
- sayed
- Tantawy

previous **Next** **Submit**

2. Has_no_question:

similar to has_more_question function. Used for previous button. If there is question before the current question, the previous button goes to the previous question. If not, the previous button does nothing.

3. Next_question:

Used to increase the number of the current question and update the question text.

```
def next_question(self):
    """Get the next question by incrementing the question number"""
    self.current_question = self.questions[self.question_no]
    self.question_no += 1
    q_text = self.current_question.question_text
    return f"Q.{self.question_no}: {q_text}"
```

4. Previous_question:

similar to next_question, used to decrease the number of the current question and update the question text.

5. Check_answer:

check the user's answer (that we pass to this function) against the correct answer and maintain the score. The score is an array of the same length of the questions, If the answer of the student of question (n) is correct we make index (n-1) of the array = 1. Else we make index (n-1) of the array = 0.

```
def check_answer(self, user_answer):
    correct_answer= self.current_question.correct_answer
    if user_answer.lower() == correct_answer.lower():
        self.score[(self.question_no-1)] = 1
```

```
    return True
else:
    self.score[(self.question_no
                 - 1)] = 0
    return False
```

6. Get_score:

to get the number of correct answers of the student by calculating the sum of 1s in the score array, the number of wrong answers of the student by subtracting the number of correct answers from the number of all questions, and the score percent by divide the sum of correct answers by the total number of questions and multiply 100.

```
def get_score(self):
    sum = 0
    for i in self.score:
        sum += i
    wrong = self.question_no - sum
    score_percent = int(sum / len(self.questions)) * 100
    return (sum, wrong, score_percent)
```

In python GUI, we need to display:

1. The title of the exam
2. The question
3. The four choices
4. Four radio buttons in front of each choice
5. A next button to go to the next question
6. A previous button to go to the previous question

We create a python file for this called "quiz_ui.py" for displaying them.

We import tkinter library for python GUI:

```
from tkinter import *
```

to display the root window and manages all the other components of the tkinter application. We can initialize the tkinter instance by assigning the variable to it, so we assign it to "root" variable then we set its dimensions, and geometry:

```
root = Tk()
root.attributes('-fullscreen', True)
root.geometry("700x350")
```

1. Title:

The title is just a text, we create a label for it by:

```
def display_title(self):
    title = Label(self.window, text="Exam model", width=47, bg="green",
fg="white", font=("arial", 30, "bold"))
    # place of the title
    title.place(x=0, y=2)
```

2. The question:

The question to be displayed can be get by pressing the next button or by pressing the previous button. So we create two functions:

- "display_question_next" that get the required question from "next_question()" that we have created before.
- "display_question_previous" that get the required question from "previous_question()" that we have created before.

```
def display_question_next(self):
    """To display the question"""
    self.quiz.next_question()
    self.canvas.itemconfig(self.question_text,
                          text=q_text)

def display_question_previous(self):
    """To display the question"""
    self.quiz.previous_question()
    self.canvas.itemconfig(self.question_text, text=q_text)
```

3. Next_button:

It's just a button that has next_btn action:

```
next_button =  
  
Button(self.window, text="Next", command=self.next_btn, width=13, bg="green",  
fg="white", font=("ariel", 25, "bold"))  
  
# placing the button on the screen  
next_button.place(x=780, y=600)
```

In next_btn action:

We get the user_answer and pass it to "check_answer" function to check the user answer and update the array. Then use "has_more_questions" function to check if there is another question. If there is, we use "display_question_next" & "display_options" to display the next question and its choices. If not, the submit button appears with "submit_btn" action.

```
def next_btn(self):  
    """To show feedback for each answer and keep checking for more questions"""  
  
    # Check if the answer is correct  
    self.quiz.check_answer(self.user_answer.get())  
  
    if self.quiz.has_more_questions():  
        # Moves to next to display next question and its options  
        self.display_question_next()  
        self.display_options()  
    else:  
        # submit button  
        submit_button = Button(self.window, text="Submit",  
command=self.submit_btn, width=13, bg="cyan", fg="white", font=("ariel", 25,  
"bold"))  
  
        # placing the button on the screen  
        submit_button.place(x=800, y=600)
```

in "submit_btn" action we just destroy the GUI:

```
def submit_btn(self):  
    # destroys the self.window  
    self.window.destroy()
```

4. previous_button:

It's just a button that has previous_btn action:

```
previous_button = Button(self.window,  
text="previous",command=self.previous_btn,width=13, bg="green", fg="white",  
font=("ariel", 25, "bold"))  
# placing the button on the screen  
previous_button.place(x=380, y=600)
```

In previous_btn action:

We get the user_answer and pass it to "check_answer" function to check the user answer and update the array. Then use "has_no_questions" function to check if there is no question before this question. If there is, we use "display_question_previous" & "display_options" to display the previous question and its choices. If not, nothing happens and the current question still displayed on the screen.

```
def previous_btn(self):  
    # Check if the answer is correct  
    self.quiz.check_answer(self.user_answer.get())  
  
    if not(self.quiz.has_no_questions()):  
        # Moves to previous to display question and its options  
        self.display_question_previous()  
        self.display_options()  
    else:  
        #nothin to do
```

5. Display_choices:

First, we create four radio buttons for the four choices and create an array called "choice_list" to contain the four radio buttons. Then we display the four buttons in their places:

```
def radio_buttons(self):  
    # initialize the list with an empty list of options
```

```
choice_list = []

# position of the first option
y_pos = 250

# adding the options to the list
while len(choice_list) < 4:

    # setting the radio button properties
    radio_btn = Radiobutton(self.window,
                           variable=self.user_answer, value="",
                           font=("ariel", 30))

    # adding the button to the list
    choice_list.append(radio_btn)

    # placing the button
    radio_btn.place(x=80, y=y_pos)

    # incrementing the y-axis position by 40
    y_pos += 40

# return the radio buttons
return choice_list
```

This function returns the choice_list which contains the four radio buttons.

We store the choice list in the variable opts:

```
self.opts = self.radio_buttons()
```

to display the four choices, firstly, we set the user_answer to None so that there is no black point in any radio button.

Then we loop over the choices and set each radio button with it's choice :

```
def display_options(self):
    """To display four options"""
    val = 0

    # deselecting
    self.user_answer.set(None)
```

```
# looping over the options to be displayed for the
# text of the radio buttons.
for option in self.quiz.current_question.choices:
    self.opts[val]['text'] = option
    self.opts[val]['value'] = option
    val += 1
```

at the end, the professor receives the name of the student, number of the student's correct answers, number of the student's wrong answers, the score precent of the student and an array containing 0s and 1s to inform the professor which questions does the student solve correctly and which questions does the student solve wrongly. i.e., if the array is [0, 0, 1, 1, 1, 0], that means the student has solved the 1st, 2nd and 6th questions wrongly and has solved 3rd, 4th and 5th questions correctly. We get this information from "get_score" function.

Exam module:

```
1 ######
2 from question_model import Question
3 from quiz_brain import QuizBrain
4 from quiz_ui import QuizInterface
5 from random import shuffle
6 import html
7 import socket
8 import base64
9 import json
10 #from quiz_data import question_data
11 #####
12 def receive_request_exam():
13     global message
14     data_cipher_encoded = socket_client.recv(1024)
15     message = json.loads(base64.b64decode(data_cipher_encoded).decode())
16     return message
17
18 def send_request_exam(name,correct,wrong,score_percent):
19     global message
20     message = {"name":name,"correct":correct,"wrong":wrong,"score_percent":score_percent}
21
22     data_to_send = json.dumps(message) # convert to json like string
23     socket_client.send(base64.b64encode(data_to_send.encode()))
24
25
26 #####
27 def Exam(name,exam_dectictionary):
28
29     question_data = exam_dectictionary
30     question_bank = []
31     for question in question_data:
32         choices = []
33         question_text = question["question"]
34         correct_answer = question["correct_answer"]
35         incorrect_answers = question["incorrect_answers"]
36         for ans in incorrect_answers:
37             choices.append(ans)
38         choices.append(correct_answer)
39         shuffle(choices)
40         new_question = Question(question_text, correct_answer, choices)
41         question_bank.append(new_question)
42
43     quiz = QuizBrain(question_bank)
44     quiz_ui = QuizInterface(quiz)
45     correct, wrong, score_percent = quiz.get_score()
46     #send_request_exam(name,correct,wrong,score_percent)
47     #print(name)
48     #print(correct)
49     #print(wrong)
50     #print(score_percent)
51     print("You've completed the quiz")
52     #print(f"Your final score was: {quiz.score}/{len(quiz.questions)}")
53     return(name,correct,wrong,score_percent,quiz.score[0:len(quiz.questions)])
54
55 #main
56 msg =[{"question": "what is your name?", "correct_answer": 'Tantawy', "incorrect_answers": ['Hossam', 'Gad', 'Saber']},
57         {"question": "How old are you?", "correct_answer": '22', "incorrect_answers": ['20', '21', '19']},
58         {"question": "where are you from?", "correct_answer": 'benha', "incorrect_answers": ['giza', 'aswan', 'cairo']},
59         {"question": "what is your job?", "correct_answer": 'still student', "incorrect_answers": ['engineer', 'doctor', 'teacher']},
60         {"question": "predicted graduated year?", "correct_answer": '2022', "incorrect_answers": ['2021', '2023', '2024']},
61         {"question": "your best friend?", "correct_answer": 'Tantawy', "incorrect_answers": ['sayed', 'saber', 'goda']}]
62 ret=Exam("hosam",msg)
```

Question:

```
1 class Question:
2     def __init__(self, question: str, correct_answer: str, choices: list):
3         self.question_text = question
4         self.correct_answer = correct_answer
5         self.choices = choices
```

Quiz brain:

```
1 class QuizBrain:
2     def __init__(self, questions):
3         self.question_no = -1
4         self.score = [0,0,0,0,0,0,0,0,0,0]
5         self.questions = questions
6         self.current_question = None
7
8     def has_more_questions(self):
9         """To check if the quiz has more questions"""
10
11     return self.question_no < len(self.questions)
12
13     def has_no_questions(self):
14         """To check if the quiz has no questions"""
15
16     return self.question_no < 2
17
18
19     def next_question(self):
20         """Get the next question by incrementing the question number"""
21
22         self.current_question = self.questions[self.question_no]
23         self.question_no += 1
24         q_text = self.current_question.question_text
25         return f"Q.{self.question_no}: {q_text}"
26
27     def previous_question(self):
28         """Get the previous question by decreasing the question number"""
29
30
31         if(self.question_no>0):
32             self.question_no -= 2
33         self.current_question = self.questions[self.question_no]
34         self.question_no += 1
35         q_text = self.current_question.question_text
36         return f"Q.{self.question_no}: {q_text}"
```

```

37
38     def check_answer(self, user_answer):
39         """Check the user's answer against the correct answer and maintain the score"""
40
41         correct_answer = self.current_question.correct_answer
42         if user_answer.lower() == correct_answer.lower():
43             self.score[(self.question_no - 1)] = 1
44             return True
45         else:
46             self.score[(self.question_no - 1)] = 0
47             return False
48
49     def get_score(self):
50         """Get the number of correct answers, wrong answers, and score percentage."""
51         sum = 0
52
53         for i in self.score:
54             sum = sum + i
55         wrong = self.question_no - sum
56         score_percent = int(sum / len(self.questions) * 100)
57         return (sum, wrong, score_percent)

```

Quiz:

```

1  from tkinter import Tk, Canvas, StringVar, Label, Radiobutton, Button, messagebox
2  from quiz_brain import QuizBrain
3
4  THEME_COLOR = "#375362"
5
6
7  class QuizInterface:
8
8      def __init__(self, quiz_brain: QuizBrain) -> None:
9          self.quiz = quiz_brain
10         self.window = Tk()
11         self.window.attributes('-fullscreen', True)
12         self.window.title("iQuiz App")
13         self.window.geometry("700x350")
14
15
16         # Display Title
17         self.display_title()
18
19         # Create a canvas for question text, and display question
20         self.canvas = Canvas(width=2000, height=250)
21         self.question_text = self.canvas.create_text(600, 100,
22                                         text="Question here",
23                                         width=2200,
24                                         fill=THEME_COLOR,
25                                         font=(
26                                             ('Ariel', 40, 'italic'))
27
28         self.canvas.grid(row=2, column=0, columnspan=2, pady=50)
29         self.display_question_next()
30         self.display_question_previous()
31
32         # Declare a StringVar to store user's answer
33         self.user_answer = StringVar()
34
35         # Display four options (radio buttons)
36         self.opts = self.radio_buttons()
37         self.display_options()

```

```
37         self.display_options()
38
39     # To show whether the answer is right or wrong
40     #self.feedback = Label(self.window, pady=10, font=("ariel", 15, "bold"))
41     #display wrong or right after each question
42     #self.feedback.place(x=300, y=380)
43
44     # Next and Quit Button
45     self.buttons()
46
47     # Mainloop
48     self.window.mainloop()
49
50     def display_title(self):
51         """To display title"""
52
53         # Title
54         title = Label(self.window, text="Exam model",
55                         width=47, bg="green", fg="white", font=("ariel", 30, "bold"))
56
57         # place of the title
58         title.place(x=0, y=2)
59
60     def display_question_next(self):
61         """To display the question"""
62
63         q_text = self.quiz.next_question()
64         self.canvas.itemconfig(self.question_text, text=q_text)
65
66     def display_question_previous(self):
67         """To display the question"""
68
69         q_text = self.quiz.previous_question()
70         self.canvas.itemconfig(self.question_text, text=q_text)
```

Graduation Project Report

7/2022

```
108     # looping over the options to be displayed for the
109     # text of the radio buttons.
110     for option in self.quiz.current_question.choices:
111         self.opts[val]['text'] = option
112         self.opts[val]['value'] = option
113         val += 1
114
115     def next_btn(self):
116         """To show feedback for each answer and keep checking for more questions"""
117
118         # Check if the answer is correct
119         self.quiz.check_answer(self.user_answer.get())
120
121         if self.quiz.has_more_questions():
122             # Moves to next to display next question and its options
123             self.display_question_next()
124             self.display_options()
125         else:
126             # if no more questions, then it displays the score
127             self.display_result()
128
129             # destroys the self.window
130             self.window.destroy()
131
132
133     def previous_btn(self):
134         """To show feedback for each answer and keep checking for more questions"""
135
136         # Check if the answer is correct
137         self.quiz.check_answer(self.user_answer.get())
138
139         if not(self.quiz.has_no_questions()):
140             # Moves to next to display next question and its options
141             self.display_question_previous()
142             self.display_options()
143         else:
144             # if no more questions, then it displays the score
145             self.display_result()
146
147             # destroys the self.window
148             self.window.destroy()
149
150     def buttons(self):
151         """To show next button and previous button"""
152
153         # The first button is the Next button to move to the
154         # next Question
155         next_button = Button(self.window, text="Next", command=self.next_btn,
156                               width=13, bg="green", fg="white", font=("ariel", 25, "bold"))
157         # palcing the button on the screen
158         next_button.place(x=780, y=600)
159
160
161         # previous button
162         previous_button = Button(self.window, text="previous", command=self.previous_btn,
163                               width=13, bg="green", fg="white", font=("ariel", 25, "bold"))
164
165         # palcing the button on the screen
166         previous_button.place(x=380, y=600)
167
168
169         # This is the second button which is used to Quit the self.window
170         quit_button = Button(self.window, text="Quit", command=self.window.destroy,
171                               width=5, bg="red", fg="white", font=("ariel", 16, " bold"))
```

```
172
173     # placing the Quit button on the screen
174     #quit_button.place(x=700, y=50)
175
176     def display_result(self):
177         """To display the result using messagebox"""
178         correct, wrong, score_percent = self.quiz.get_score()
179
180         correct = f"Correct: {correct}"
181         wrong = f"Wrong: {wrong}"
182
183         # calculates the percentage of correct answers
184         result = f"Score: {score_percent}%"
185
186         # Shows a message box to display the result
187         # messagebox.showinfo("Result", f"{result}\n{correct}\n{wrong}")
```

3.8 Facial recognition:

This function is used to open the picamera module to recognize the person using raspberry pi.

In our project we use this function for two reasons:

1. In lectures when the professor wants to take the absence of students, the camera of the raspberry pi open to check who is present.
2. Is there is an exam the camera raspberry pi open to check the required student is using the raspberry pi or there is another one? If there another one the camera check if the student is another student in the class or it's an unknown person.

First, we use videostream function to initialize the video stream and allow the camera sensor to warm up. The same function can be used to use USB camera or picamera module. Depending on the parameters supplied to the VideoStream constructor, the appropriate video stream class (either for the USB camera or picamera module) will be instantiated.

In our code we use picamera module, so we initialize picamera by:

```
vs = VideoStream(usePiCamera=True).start()
```

After executing `VideoStream(usePiCamera=True).start()` we need to halt the flow of the program so that several other executions can take place or simply due to the utility required. `sleep()` can come handy in such a situation which provides an accurate and flexible way to halt the flow of code for any period of time.

We use `time.sleep(2.0)` for this purpose.

Now the live video is running. To check who is using raspberry pi, we take a frame of a video by:

```
frame = vs.read()
```

then detect the position of the faces in the frame by:

```
boxes = face_recognition.face_locations(frame)
```

then we encode the part of the frame containing the face as mentioned in training model by:

```
encodings = face_recognition.face_encodings(frame, boxes)
```

After that we compare between the encoding of the face of the student using the raspberry pi and the encodings of our data base by:

```
matches = face_recognition.compare_faces(data["encodings"], encoding)
```

encoding: is between the encoding of the face of the student using the raspberry pi.

data["encodings"]: are the encodings of our data base.

If the student is in the dataset the function returns his name. If not the function return "unknown".

Face recognition module:

```
1  #! /usr/bin/python
2
3  ##prototypes of file##
4  #recognize_student()
5  #get_student_name():
6
7
8  # import the necessary packages
9  from imutils.video import VideoStream
10 from imutils.video import FPS
11 import face_recognition
12 import imutils
13 import pickle
14 import time
15 import cv2
16 import webbrowser
17 student_name=("none")
18 def recognize_student(student_name_from_server):
19     #global counts
20     timer=0
21     student_name=("none")
22     first_doing = 0
23     last_name=("none")
24     last_last_name=("none")
25     counter= 0
26     #Initialize 'currentname' to trigger only when a new person is identified.
27     currentname = "unknown"
28     #Determine faces from encodings.pickle file model created from train_model.py
29     encodingsP = "encodings.pickle"
30
31     # load the known faces and embeddings along with OpenCV's Haar
32     # cascade for face detection
33     print("[INFO] loading encodings + face detector...")
34     data = pickle.loads(open(encodingsP, "rb").read())
35
36     # initialize the video stream and allow the camera sensor to warm up
37     # Set the ser to the followng
38     # src = 0 : for the build in single web cam, could be your laptop webcam
```

Graduation Project Report

7/2022

```
39      # src = 2 : I had to set it to 2 inorder to use the USB webcam attached to my laptop
40      #vs = VideoStream(src=2,framerate=10).start()
41      vs = VideoStream(usePiCamera=True).start()
42      time.sleep(2.0)
43
44      # start the FPS counter
45      fps = FPS().start()
46
47      # loop over frames from the video file stream
48      while True:
49          currentname = "unknown"
50          # grab the frame from the threaded video stream and resize it
51          # to 500px (to speedup processing)
52          frame = vs.read()
53          #frame = imutils.resize(frame, width=500)
54          # Detect the face boxes
55          boxes = face_recognition.face_locations(frame)
56          # compute the facial embeddings for each face bounding box
57          encodings = face_recognition.face_encodings(frame, boxes)
58          names = []
59
60          # loop over the facial embeddings
61          for encoding in encodings:
62              # attempt to match each face in the input image to our known
63              # encodings
64              matches = face_recognition.compare_faces(data["encodings"],
65                  encoding)
66              name = "Unknown" #if face is not recognized, then print Unknown
67
68
69              # check to see if we have found a match
70              if True in matches:
71                  # find the indexes of all matched faces then initialize a
72                  # dictionary to count the total number of times each face
73                  # was matched
74                  matchedIdxs = [i for (i, b) in enumerate(matches) if b]
75                  counts = {}
76
77
78                  # loop over the matched indexes and maintain a count for
79                  # each recognized face face
80                  for i in matchedIdxs:
81                      name = data["names"][i]
82                      counts[name] = counts.get(name, 0) + 1
83
84
85                  # determine the recognized face with the largest number
86                  # of votes (note: in the event of an unlikely tie Python
87                  # will select first entry in the dictionary)
88                  name = max(counts, key=counts.get)
89                  timer +=1
90                  print(counts[name])
91                  #If someone in your dataset is identified, print their name on the screen
92                  if currentname != name :
93                      if counts[name]>600:
94                          currentname = name
95                          print(currentname)
96                          print(counts[name])
97                          names.append(currentname)
98                      else :
99                          print(currentname)
100                         print(counts[name])
101                         names.append(currentname)
102
103                         print(currentname)
104                         print(counts[name])
105                         names.append(currentname)
```

Graduation Project Report

7/2022

```
104
105     # update the list of names
106
107     #name = "Unknown"
108     # loop over the recognized faces
109     for ((top, right, bottom, left), name) in zip(boxes, names):
110         # draw the predicted face name on the image - color is in BGR
111         cv2.rectangle(frame, (left, top), (right, bottom),
112                         (0, 255, 225), 2)
113         y = top - 15 if top - 15 > 15 else top + 15
114         cv2.putText(frame, name, (left, y), cv2.FONT_HERSHEY_SIMPLEX,
115                     .8, (0, 255, 225), 2)
116
117     # display the image to our screen
118     cv2.imshow("Facial Recognition is Running", frame)
119     key = cv2.waitKey(1) & 0xFF
120
121     # quit when 'q' key is pressed
122     if currentname == ("Mohammed"):
123         #
124         counter = counter + 1
125         #
126         if counter == 5:
127             #
128             webbrowser.open("https://docs.google.com/forms/d/e/1FAIpQLSeo1B0shXPHVNZEBY8uagL0DN9xP6M0LFbR8sKYyXP8z:")
129             #
130             break
131
132             if first_doing == 0:
133                 last_name = currentname
134             elif first_doing == 1:
135                 last_last_name = last_name
136                 last_name = currentname
137                 print("currentname : {}".format(currentname))
138                 print("last_name : {}".format(last_name))
139                 print("last_last_name : {}".format(last_last_name))
140                 #print("counts: {}".format(counts[name]))
141             elif first_doing == 2:
142                 first_doing = 0
143                 if (currentname == last_name)and(currentname == last_last_name)and(currentname == student_name_from_server):
144                     counter = counter + 1
```

```
141
142     print("-currentname : {}".format(currentname))
143     print("-last_name : {}".format(last_name))
144     print("-counter : {}".format(counter))
145     print("-last_last_name : {}".format(last_last_name))
146     # print("counts: {}".format(counts[name]))
147     if counter == 2:
148         student_name_from_recognition = currentname
149         break
150     first_doing = first_doing + 1
151     if(timer==20):
152         student_name_from_recognition = currentname
153         break
154
155     if key == ord("q"):
156         break
157
158     #if curname
159     # update the FPS counter
160     fps.update()
161
162     # stop the timer and display FPS information
163     fps.stop()
164     print("[INFO] elapsed time: {:.2f}".format(fps.elapsed()))
165     print("[INFO] approx. FPS: {:.2f}".format(fps.fps()))
166
167     # do a bit of cleanup
168     cv2.destroyAllWindows()
169     vs.stop()
170     print("student name in recognize student func: {}".format(student_name))
171     return student_name_from_recognition
172
173 print(recognize_student("Mohammed"))|
```

3.9 Train model:

After the administration had taken some pictures to the student, which now saved in the folder "New" each student in his folder, these pictures need to be encoded so that any student has a code represents his face which differs from any other faces. This function is used to this purpose.

First, we enter the folder "New" which contains some of folders each has a name of a student and contains the student's pictures.

We enter the folders, folder by folder by:

```
img_old_path = "New/"  
imagePaths=list(paths.list_images(img_old_path))  
for(i,imagePath)in enumerate(imagePaths):
```

we store the name of the folder,which is the name of the student, in a variable called name by:

```
name = imagePath.split(os.path.sep)[-2]
```

Now the folder containing some of pictures of the same student we read them picture by picture by:

```
image = cv2.imread(imagePath)
```

the important part in the picture is the face of the student, so we firstly get it from the picture to encode it by:

```
boxes = face_recognition.face_locations(rgb,model="hog")  
# compute the facial embedding for the face  
encodings = face_recognition.face_encodings(rgb, boxes)
```

then we create a pickle file containing two arrays.

One array contains the names of the student and the second one is containing the corresponding encodes by:

```
knownEncodings.append(encoding)  
knownNames.append(name)
```

```
data = {"encodings": knownEncodings, "names": knownNames}
f = open("encodings.pickle", "wb")
f.write(pickle.dumps(data))
f.close()
```

Train model:

```
1  #!/usr/bin/python
2
3  # import the necessary packages
4  from imutils import paths
5  import face_recognition
6  #import argparse
7  import pickle
8  import cv2
9  import os
10 import shutil
11 def start_train():
12     #req_name=input("enter the name for train model: ")
13     # our images are located in the dataset folder
14     print("[INFO] start processing faces...")
15     img_old_path = "New/"
16     imagePaths = list(paths.list_images(img_old_path))
17
18     # initialize the list of known encodings and known names
19     encodingsP = "encodings.pickle"
20     data = pickle.loads(open(encodingsP, "rb").read())
21     knownEncodings = data["encodings"]
22     knownNames = data["names"]
23     #knownEncodings = []
24     #knownNames = []
25     # loop over the image paths
26     for (i, imagePath) in enumerate(imagePaths):
27         # extract the person name from the image path
28
29         print("[INFO] processing image {}/{}".format(i + 1,
30             len(imagePaths)))
31         name = imagePath.split(os.path.sep)[-2]
32
33         # load the input image and convert it from BGR (OpenCV ordering)
34         image = cv2.imread(imagePath)
35
36         #adding image to dataset with its folder name
37         img_path_ = "dataset/" + name
```

```
37     img_path_ = "dataset/" + name
38     if not os.path.exists(img_path_):
39         os.mkdir(img_path_)
40     img_name_ = img_path_ + "/image_{}.jpg".format(i)
41     cv2.imwrite(img_name_, image)
42
43     # to dlib ordering (RGB)
44     rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
45
46     # detect the (x, y)-coordinates of the bounding boxes
47     # corresponding to each face in the input image
48     boxes = face_recognition.face_locations(rgb,
49         model="hog")
50
51     # compute the facial embedding for the face
52     encodings = face_recognition.face_encodings(rgb, boxes)
53
54     # loop over the encodings
55     for encoding in encodings:
56         # add each encoding + name to our set of known names and
57         # encodings
58         knownEncodings.append(encoding)
59         knownNames.append(name)
60
61     #delete the old folder called New and create ather one with same name
62     shutil.rmtree(img_old_path)
63     os.mkdir(img_old_path)
64
65     # dump the facial encodings + names to disk
66     print("[INFO] serializing encodings...")
67     data = {"encodings": knownEncodings, "names": knownNames}
68     f = open("encodings.pickle", "wb")
69     f.write(pickle.dumps(data))
70     f.close()
71
72     #start_train()
```

3.10 Video pictures:

When the student want to assign a class, the administrator use this function to take some pictures to the student.

A video stream is taken to the student and the number of pictures, determined by the administrator, is taken from this video.

First, we write the parameters of videos to select resolution and number of frames in second:

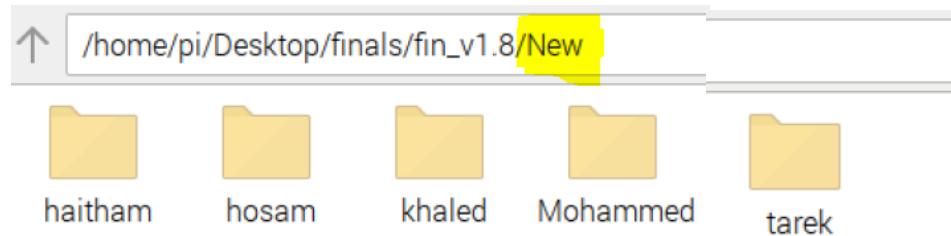
```
cam = PiCamera()  
cam.resolution = (512, 304)  
cam framerate = 10
```

There is a main folder called "New" in which these pictures are automatically saved in a new folder called the name of the student, determined by the administrator by:

```
folder_name = "New/" + name  
if not os.path.exists(folder_name):  
    os.mkdir(folder_name)
```

for example: If the administrator runs the program for students:

"haitham", "Mohammed", "Tarek" this would be the output:

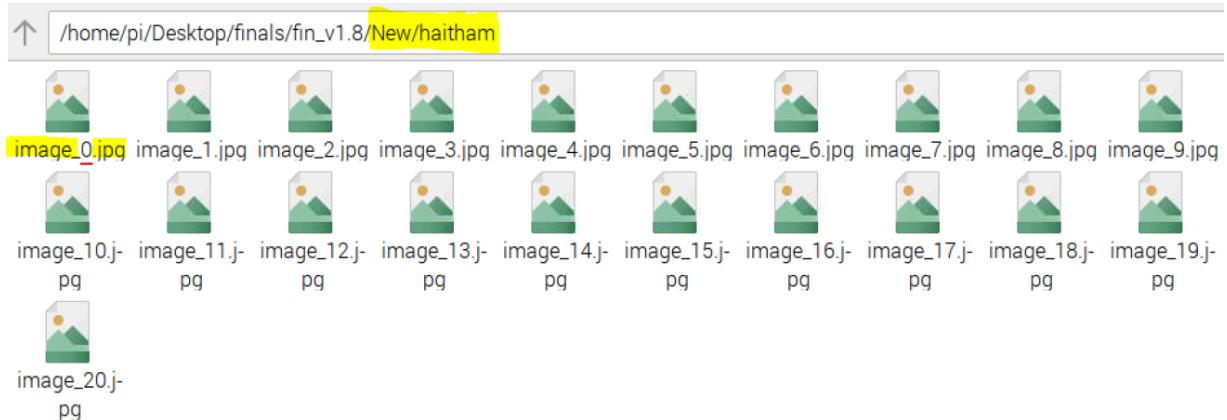


The pictures are called "image_(number of picture)" and are stored in New folder in the student folder by the following:

```
if img_counter <= no_pictures:  
    img_name = "New/" + name + "/image_{ }.jpg".format(img_counter)  
    cv2.imwrite(img_name, image)
```

```
print("{"  
    img_counter += 1  
written!" .format(img_name))
```

like this:



This function also creates a pickle file containing two arrays.

One array containing the names of the students. And the other one containing the corresponding IDs of these students.

We first store the two arrays in a variable called data by:

```
data={"names": knownNames,"IDs":knownIds}  
knownNames.append(the_name)  
knownIds.append(student_ID)
```

Then we dump the data to the pickle file by:

```
f = open("name_id.pickle", "wb")  
f.write(pickle.dumps(data))  
f.close()
```

these are the two arrays in the "name_id.pickle" file:

```
['hosam', 'tarek', 'Mohammed', 'khaled' 'haitham']  
[20, 150, 200, 50, 2000]
```

```
1 import cv2
2 from picamera import PiCamera
3 from picamera.array import PiRGBArray
4 import os
5 import pickle
6
7 def get_pictures(the_name,no_pictures,student_ID):
8     Name_ID_P = "name_id.pickle"
9     data = pickle.loads(open(Name_ID_P, "rb").read())
10    knownNames=data["names"]
11    knownIds=data["IDs"]
12    data = {"names": knownNames, "IDs":knownIds}
13    exist=0
14    count = 0
15    for y in range(len(knownNames)):
16        if (data["names"][y] == the_name):
17            exist=1
18            if (count > 0):
19                data["names"][y]=""
20                data["IDs"][y]=""
21            if (count == 0):
22                count=count+1
23                data["IDs"][y]=student_ID
24
25
26    if (exist==0):
27        knownNames.append(the_name)
28        knownIds.append(student_ID)
29    for y in range(len(knownNames)):
30        print(data["names"][y])
31        print(data["IDs"][y])
32    f = open("name_id.pickle", "wb")
33    f.write(pickle.dumps(data))
34    f.close()
35
36    #for i in data["names"]:
37    #    print(i)
38    #for i in data["IDs"]:
```

```
64         if img_counter <= no_pictures:
65             img_name = "New/" + name + "/image_{}.jpg".format(img_counter)
66             cv2.imwrite(img_name, image)
67             print("{} written!".format(img_name))
68             img_counter += 1
69
70
71         if k%256 == 27: # ESC pressed
72             break
73         elif k%256 == 32:
74             # SPACE pressed
75             img_name = "New/" + name + "/image_{}.jpg".format(img_counter)
76             cv2.imwrite(img_name, image)
77             print("{} written!".format(img_name))
78             img_counter += 1
79
80         if k%256 == 27:
81             print("Escape hit, closing...")
82             break
83
84     cv2.destroyAllWindows()
85
86 #main
87 #get_pictures("Mohammed",20,200)
88 #print("welcome here to start taking pictures use 'get_pictures(the_name,no_frames,no_pictures)'")|
```

3.11 Voting:

During the lecture, if the professor wants to create a vote, he uses this function.

The professor writes the question, and the choices then send them to student and receive from the student a variable containing the choice the student has chosen

Using the python GUI, we represent this information for student.

We import tkinter library for python GUI:

```
from tkinter import *
```

to display the root window and manages all the other components of the tkinter application. We can initialize the tkinter instance by assigning the variable to it, so we assign it to "root" variable then we set its dimensions, title and geometry:

```
root = Tk()
root.attributes('-fullscreen', True)
root.title("Vote App")
root.geometry("700x350")
```

now we have 7 pieces of information to represent for the student:

1. The question
- 2-5. The four choices
6. Submit button
7. A label to inform the student the choose he have selected



1. The question is just a text. We create a label for it by:

```
label_question = Label(root, text=choices["question"], font=('Courier 44 underline'))
label_question.pack()
```

7. The last label is just a text containing the choice of the student. At first it has a text "you choose: nothing" until the student press any choice. We create a label for it by:

```
label_choice = Label(root, textvariable=choose, font=('Courier 44 underline'))
choose.set("you choose: nothing")
label_choice.pack()
```

2-5. the four choices are buttons has an action to change the variable containing the choice the student has chosen by:

```
button1 = Button(master = root, text=choices["first"],command=lambda *args:  
change_choice(choices["first"]),width=130, bg="green", fg="white", font=("ariel",  
40, "bold")).pack()
```

the action of each button of choices call change_choice function which change the value of the variable containing the choice of the student and update the last label by the choice of the student as following:

```
def choose : "  
    val="you" choose : "  
    choose.set(val)  
    global variable =  
    variable value  
    print (variable)  
change_choice(value):  
    + value
```

6. the submit button is a button has an action to destroy the root tkinter as following:

```
submit_button = Button(master = root,  
text='Submit',command=root.destroy,width=80, bg="yellow", fg="black",  
font=("ariel", 40, "bold"))  
submit_button.pack(pady=20)
```

At the end the function returns the choice of the student.

Graduation Project Report

7/2022

```
1  from tkinter import *
2  variable = "no answer"
3  def change_choice(value):
4      val="you choose : " + value
5      choose.set(val)
6      global variable
7      variable = value
8      print (variable)
9  def voting(choices):
10     global label_choice
11     global label_question
12     global button1
13     global button2
14     global button3
15     global button4
16     global choose
17     global root
18     root = Tk()
19     root.attributes('-fullscreen', True)
20     root.title("Vote App")
21     root.geometry("700x350")
22
23     choose= StringVar()
24
```

```
37     label_choice.pack()
38
39     root.mainloop()
40
41     return variable
42
43     #...etc
44     #main
45     msg={
46         "question":"what is your best color?",
47         "first":"red",
48         "second":"blue",
49         "third":"black",
50         "fourth":"white"
51     }
52     voting(msg)
```

3.12 Conclusion

The Raspberry Pi is a powerful little beast and a great platform for building low-cost, but highly capable, embedded systems. The interfaces built into its GPIO connector make it easy to bolt on modules using simple low-cost electronics and a bit of configuration to create very functional and flexible systems. The inclusion of a dedicated camera interface and networking interfaces give you everything you could possibly need for an Internet-connected home security system.

4. Chapter Four: Web Application front end

4.1 Introduction

In our project, we tried to make the design as simple as possible. The simplicity of the design is related to web pages as most people in the world are currently very familiar with web pages in the internet. While swimming in the web pages implementation, we will dive into html.

4.2 why using html?

Actually, HTML has many advantages to use in our webpage design

(1) HTML is free

One of the biggest advantages of HTML is that it is free of cost, and there is no need to purchase specific software. One should not have to deal with different plugins required to work on any software as HTML does not require any plugins. So it is very cost-effective from a per business perspective as there is no cost of purchasing the license if the whole website is developed in HTML language.

(2) HTML is supported by all browsers

HTML supports almost all browsers around the globe. So there is no need to worry about the website written in HTML for the browser support as the website would easily show up in all the browsers if the program keeps in mind to optimize the website for the different browsers. HTML provides an easy way to optimize the website in HTML according to browsers to the web developers.

(3) HTML is the most friendly search engine

HTML is one of the most friendly search engines in comparison to all the programming languages available in the market (Search Engine friendly means delivering users quality websites with relevant information when searched for a particular one). It is quite easier to create SEO compliant websites using HTML than other programming languages. HTML websites are easier to read and accessed by web crawlers and hence reduces parsing time and the page load time of the website hence improving its performance.

(4) HTML is simple to edit

HTML is very easy to edit as there is no need to have a special interface or platform to edit it. It is written in simple Notepad and hence can be simply edited in any text editor like Notepad, Notepad++, etc.

(5) HTML can integrate easily with other languages

HTML can be easily integrated with multiple languages and does not create any issues in it. For example, in Javascript, Php, node.js, CSS and many more, we write the code of these languages between the HTML, and it mixes with them very easily.

(6) HTML is lightweight

HTML is lightweight language. It has a high signal to noise ratio as compared to other forms of communication. It is also faster to download HTML code, which means it is highly compressive also.

(7) HTML is basic for all programming languages

For the programmer to be either a frontend or backend developer, one must have knowledge of HTML as it is the basic language and all the other languages integrate with it while coding like JavaScript, JSP, Php, etc. Similarly, XML syntax is just like HTML and XML, which is used these days widely for data storage. If one has good knowledge of HTML, it is easy working with XML too for him.

(8) HTML is user-Friendly

HTML is a user-friendly programming language. One does not need to have any prior knowledge of any language. Understanding simple English is sufficient to work with it.

HTML is used in frontend development for over so many years before we have no other languages available in the market for web development. Although HTML provides all the tags to the user to add everything in the webpage like a table, images, hyperlink, etc. there were some drawbacks which were covered in the latest version of HTML, i.e. HTML5, which allows the user to insert a graphic, multimedia, semantic elements to develop powerful websites and improving UX consistently.

All these advantages will of course attract us to use HTML to design a webpage available for students, professors, admins or even guests to access our page. As in our project we access every think locally, guests can't access any information about the organization and so we can't show to the public the reasons to choose our organization.

This made us use HTML to make a design available to any kind of people and of course each kind has specific authorization.

4.3 The webpages made in our project

(1) The login page in the raspberry pi

This page includes a text input for the student to write his/her name and a button to click after writing the name. It's available only for the students.

This helps the raspberry pi to check this person without consumption a lot of power. When the student write his name, the field that the raspberry pi will search for this student's pictures only not all pictures of all students which make the search quicker without a lot of power consumption.

The screenshot shows a simple login interface. At the top, there is a solid black bar with the word "LOG IN" centered in white capital letters. Below this, there is a light orange rectangular input field with the label "NAME:" in bold black text. Inside the input field, the placeholder text "Enter your name" is visible in a smaller, lighter font. At the bottom of the page, there is a light orange rectangular button with the word "SUBMIT" centered in black capital letters.

After pressing submit, the student is forwarded to a waiting page with the remaining time to the next lecture or the next exam which we call it “timer page”.

❖ Our brilliant code for this page :

```
<head>
    <title>LOGIN</title>
    <link rel="stylesheet" type="text/css" href="login.css">
```

```
<link rel="stylesheet" type="text/css" href="style.css">

</head>

<body>

    <header>
        <h1>LOG IN</h1>
    </header>

    <main>

        <form      id="login_form"      class="form_class"      action="login.php"
method="post">

            <div class="form_div">

                <p id="no1">Name:</p>

                <input  class="field_class" ,  use-keyboard-input"  name="user_id"
type="text" placeholder="Enter your name">

                <button  class="submit_class"  type="submit"  form="login_form"
onclick="">Submit</button>

            </div>

        </form>

    </main>

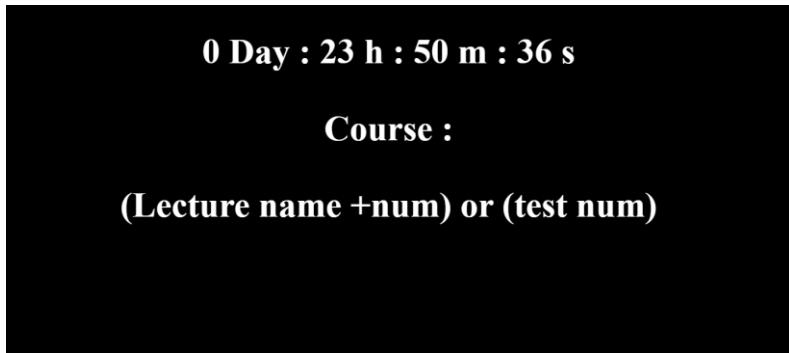
    <script src="keyboard.js"></script>

</body>
```

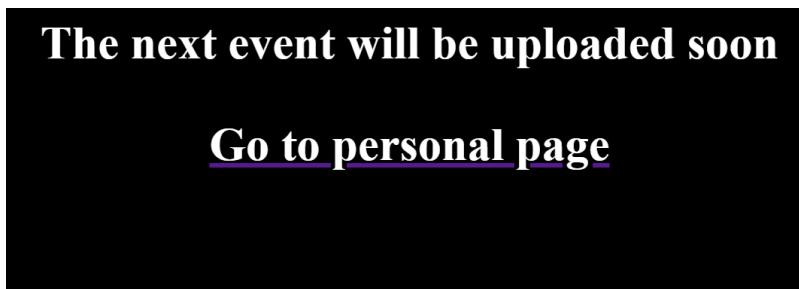
(2) Timer page

This page includes three information displayed for the student:

- The remaining time for the next event (Lecture or Exam)
- The name of the course related to the next event
- Information about the event.



In case the professor haven't uploaded the next event yet, the student will see the page shown in the picture below:



So, if there is no event, the student is restricted to access his/her personal page, we will take about it below, which includes information him/her and the courses he/she studies.

❖ Our brilliant code for this page :

```
<!DOCTYPE HTML>

<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Waiting</title>
<style>
body {
background: black;
```

```
}

p {
    text-align: center;
    font-size: 40px;
    margin-top: 40px;
    color: white;
    font-weight: bold;
}

</style>
</head>
<body>

<p id="demo"></p>
<p id="current_course" class="h6" >Course :      </p>
<p id="current_action" class="h6" >(Lecture name +num) or (test num) </p>
<a href="Profile page/SP.html" style="display: none;" id="profile"><p class="h6">
> Go to personal page </p></a>
<script>

// Set the date we're counting down to
var countDownDate = new Date("July 14, 2022 15:04:40").getTime();

// Update the count down every 1 second
var x = setInterval(function() {

    // Get today's date and time
    var now = new Date().getTime();
    var distance = countDownDate - now;
    var days = Math.floor(distance / (1000 * 60 * 60 * 24));
    var hours = Math.floor((distance % (1000 * 60 * 60 * 24)) / (1000 * 60 * 60));
    var minutes = Math.floor((distance % (1000 * 60 * 60)) / (1000 * 60));
    var seconds = Math.floor((distance % (1000 * 60)) / 1000);

    document.getElementById("demo").innerHTML = days + "d " + hours + "h " + minutes + "m " + seconds + "s ";

    if (distance < 0) {
        clearInterval(x);
        document.getElementById("demo").innerHTML = "EXPIRED";
    }
})
```

```
var now = new Date().getTime();

// Find the distance between now and the count down date
var distance = countDownDate - now;

// Time calculations for days, hours, minutes and seconds
var days = Math.floor(distance / (1000 * 60 * 60 * 24));
var hours = Math.floor((distance % (1000 * 60 * 60 * 24)) / (1000 * 60 * 60));
var minutes = Math.floor((distance % (1000 * 60 * 60)) / (1000 * 60));
var seconds = Math.floor((distance % (1000 * 60)) / 1000);

// Output the result in an element with id="demo"
document.getElementById("demo").innerHTML = days + " Day : " + hours + " h
: "
+ minutes + " m : " + seconds + " s ";

// If the count down is over, write some text
if (distance < 0) {
    clearInterval(x);
    document.getElementById("demo").innerHTML = "The next event will be
uploaded soon";
    document.getElementById("current_course").style.display = "none";
    document.getElementById("current_action").style.display = "none";
    document.getElementById("profile").style.display = "block";
}
}, 1000);
</script>
```

```
</body>  
</html>
```

(3) Student personal page

This page includes the student personal information that is registered in the database, information about the courses the student studies and the grades he has taken in these courses.

❖ Our brilliant code for this page :

```
<!DOCTYPE html>  
  
<html lang="en-US">  
  
<head>  
  
    <meta charset="UTF-8">  
  
    <meta http-equiv="X-UA-Compatible" content="IE=edge">  
  
    <meta name="viewport" content="width=device-width, initial-scale=1">  
  
    <title>Profile page</title>  
  
    <link href="https://fonts.googleapis.com/css?family=Montserrat:400,700,200" rel="stylesheet">  
  
    <link href="https://maxcdn.bootstrapcdn.com/font-awesome/latest/css/font-awesome.min.css" rel="stylesheet">  
  
    <link href="css/aos.css?ver=1.1.0" rel="stylesheet">  
  
    <link href="css/bootstrap.min.css?ver=1.1.0" rel="stylesheet">  
  
    <link href="css/main.css?ver=1.1.0" rel="stylesheet">  
  
<noscript>  
  
    <style type="text/css">  
  
        [data-aos] {  
  
            opacity: 1 !important;
```

```
        transform: translate(0) scale(1) !important;  
    }  
  
    </style>  
  </noscript>  
</head>  
  
<body id="top">  
  
<header>  
  <div class="profile-page sidebar-collapse">  
    <nav class="navbar navbar-expand-lg fixed-top navbar-transparent bg-primary" color-on-scroll="400">  
      <div class="container">  
        <div class="navbar-translate"><a class="navbar-brand" href="/student-profile" rel="tooltip" id="pic"></a></div>  
        <div class="collapse navbar-collapse justify-content-end" id="navigation">  
          <ul class="navbar-nav">  
            <li class="nav-item"><a class="nav-link" href="/stdlog" smooth-scroll>Home</a></li>  
            <li class="nav-item"><a class="nav-link" href="#about" smooth-scroll>About</a></li>  
            <li class="nav-item"><a class="nav-link" href="#experience" smooth-scroll>Courses</a></li>  
            <li class="nav-item"><a class="nav-link" href="/logout/students" smooth-scroll>Logout</a></li>  
          </ul>  
        </div>  
      </div>  
    </nav>  
  </div>  
</body>
```

```
</div>
</div>
</nav>
</div>

</header>
<div class="page-content">
<div>
<div class="profile-page">
<div class="wrapper">
<div class="page-header page-header-small" filter-color="green">
<div class="page-header-image" data-parallax="true"
style="background-image: url('images/project-1.jpg')"></div>
<div class="container">
<div class="content-center">
<div class="cc-profile-image"><a href="#"></a></div>
<div class="h2 title">
{ {Auth::guard('students')->user()->first_name} }
{ {Auth::guard('students')->user()->second_name} }
@php($user = Auth::guard('students')->user())
{ {-- {$user} --} }
</div>
</div>
</div>
</div>
```

```
</div>
</div>
<div class="section" id="about">
    <div class="container">
        <div class="card" data-aos="fade-up" data-aos-offset="10">
            <div class="row">
                <div class="col-lg-6 col-md-12">
                    <div class="card-body">
                        <div class="h4 mt-0 title">Basic Information</div>
                        <div class="row">
                            <div class="col-sm-4"><strong>Name:</strong></div>
                                <div class="col-sm-8">{$user->first_name . " ". $user->second_name . " ". $user->third_name . " ". $user->fourth_name}</div>
                        </div>
                        <div class="row mt-3">
                            <div class="col-sm-4"><strong>Year:</strong></div>
                            <div class="col-sm-8">{$user->current_year}</div>
                        </div>
                        <div class="row mt-3">
                            <div class="col-sm-4"><strong>Department:</strong></div>
                            <div class="col-sm-8">[From database]</div>
                        </div>
                        <div class="row mt-3">
```

```
<div class="col-sm-4"><strong class="text-  
uppercase">Email:</strong></div>  
  
<div class="col-sm-8">{$user->email}</div>  
  
</div>  
  
</div>  
  
</div>  
  
</div>  
  
</div>  
  
</div>  
  
</div>  
  
<div class="section" id="experience">  
  
<div class="container cc-experience">  
  
<div class="h4 text-center mb-4 title">Courses</div>  
  
{-- start card--}  
  
<div class="card">  
  
<div class="row">  
  
<div class="col-md-3 bg-primary" data-aos="fade-right" data-aos-  
offset="50" data-aos-duration="500">  
  
<div class="card-body cc-experience-header">  
  
<div class="h5">[Name of the course1 from database]</div>  
  
</div>  
  
</div>  
  
<div class="col-md-9" data-aos="fade-left" data-aos-offset="50"  
data-aos-duration="500">  
  
<div class="card-body">
```

```
<p>[Discription of the course from database (added by admin)]</p>

<p>Number of bonuses: </p>
<p>Number of minuses: </p>
<p>Quizes grades: </p>
<p>Final test grade: </p>
<p>Total grade: </p>

</div>
</div>
</div>
</div>
<!-- end card--> }

</div>
</div>
</div>
</div>

<footer class="footer">
    <div class="h4 title text-center">Benha Faculty of Engineering</div>
    <div class="text-center text-muted">
        <p>&copy; COPYRIGHT © 2022 BENHA FACULTY OF ENGINEERING ALL RIGHTS RESERVED.<br>Design - <a class="credit" href="https://www.linkedin.com/in/mahmoud-gouda-409382172/" target="_blank">Mahmoud Gouda</a></p>
    </div>
</footer>
```

```
<script src="js/core/jquery.3.2.1.min.js?ver=1.1.0"></script>
<script src="js/core/popper.min.js?ver=1.1.0"></script>
<script src="js/core/bootstrap.min.js?ver=1.1.0"></script>
<script src="js/now-ui-kit.js?ver=1.1.0"></script>
<script src="js/aos.js?ver=1.1.0"></script>
<script src="scripts/main.js?ver=1.1.0"></script>
</body>
</html>
```

(4) Main login page

This page is available locally or in the internet for the students, professors and admins to login.

- *For professor:* Logging in directs the professor to the main page of this professor which allow him to upload the next event for the students whether it is a lecture or an exam.
- *For admin:* Logging in directs the admin the main page of the admin which allow him to add or edit courses, students and professors.
- *For student:* Logging in directs the student to the personal page of the student.

❖ Our brilliant code for this page :

```
<head>
  <title>LOGIN</title>
  <link rel="stylesheet" type="text/css" href="login.css">
  <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
```

```
<header>
    <h1>BENHA FACULTY OF ENGINEERING</h1>
</header>
<main>
    <form id="login_form" class="form_class" action="login" method="post">
        @csrf
        <p id="log">LOGIN AS</p>
        <input name="role" value="STUDENT" class="field_class , use-keyboard-input" list="list1" placeholder="Who are you?">
        <datalist id="list1">
            <option value="STUDENT"></option>
            <option value="PROFESSOR"></option>
            <option value="ADMIN"></option>
        </datalist>
        @error('role')
        {{ $message }}
        @enderror
        <br>
        <div class="form_div">
            <p id="no1">ID:</p>
            <input name="id" value="47" class="field_class , use-keyboard-input" type="text" placeholder="Enter your ID" maxlength="20">
            @error('id')
            {{ $message }}
            @enderror
            <p id="no2">PASSWORD:</p>
```

```
<input name="password" value="password" id="pass" class="field_class ,  
use-keyboard-input" name="password_txt" type="password" placeholder="Enter  
the password" maxlength="20">  
  
    @error('password')  
        {{$message}}  
    @enderror  
  
    <button class="submit_class" type="submit"  
form="login_form">Submit</button>  
  
    </div>  
  
</form>  
</main>  
  
<footer>  
    <br><br>  
    <p>Designed by: <a href="https://www.linkedin.com/in/mahmoud-gouda-  
409382172/" target ="_blank">Mahmoud Gouda</a></p>  
</footer>  
</body>
```

(5) Professor main page

This page includes everything the professor can use to schedule a lecture by adding its information or an exam by adding all questions and answers.

❖ Our brilliant code for this page :

```
<!DOCTYPE html>
```

```
<html lang="en">

<head>

    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="TemplateMo">
    <link href="https://fonts.googleapis.com/css?family=Poppins:100,200,300,400,500,600,700,800,900" rel="stylesheet">
    <title>Home Page</title>

    <!-- Bootstrap core CSS -->
    <link href="vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
    <!-- Additional CSS Files -->
    <link rel="stylesheet" href="assets/css/fontawesome.css">
    <link rel="stylesheet" href="assets/css/templatemo-edu-meeting.css">
    <link rel="stylesheet" href="assets/css/owl.css">
    <link rel="stylesheet" href="assets/css/lightbox.css">
    <link rel="stylesheet" type="text/css" href="login.css">
    <link rel="stylesheet" type="text/css" href="style.css">
```

```
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"></sc
ript>

</head>

<body>

<!-- Sub Header -->
<div class="sub-header">
<div class="container">
<div class="row">
<div class="col-lg-8 col-sm-8">
<div class="left-content">
<p>Welcome to <em>Benha University</em></p>
</div>
</div>
</div>
</div>

<!-- ***** Header Area Start ***** -->
<header class="header-area header-sticky">
<div class="container">
<div class="row">
<div class="col-12">
```

```
<nav class="main-nav">  
    <!-- ***** Logo Start ***** -->  
    <br> <br>  
    <a href="/profLog" class="logo">  
        Benha Faculty of Engineering  
    </a>  
    <!-- ***** Logo End ***** -->  
    <!-- ***** Menu Start ***** -->  
    <ul class="nav">  
        <li><a href="#about"></a></li>  
        <li class="scroll-to-section"><a href="/pp"></a></li>  
    </ul>  
    <a class='menu-trigger'>  
        <span>Menu</span>  
    </a>  
    <!-- ***** Menu End ***** -->  
    </nav>  
    </div>  
    </div>  
    </div>  
</header>  
<!-- ***** Header Area End ***** -->  
<!-- Model -->
```

```
<!-- Model -->
<!-- ***** Main Banner Area Start ***** -->
<section class="section main-banner" id="top" data-section="section1">
  <video autoplay muted loop id="bg-video">
    <source src="assets/images/vv.mp4" type="video/mp4" />
  </video>
  <div class="video-overlay header-text">
    <div class="container">
      <div class="row">
        <div class="col-lg-12">
          <div class="caption">
            <h6>Hello Prof. [get the name from database]</h6>
            <h2>Welcome to Benha Faculty Of Engineering</h2>
          </div>
        </div>
      </div>
    </div>
  </div>
  <br> <br> <br> <br> <br> <br> <br> <br>
</section>
<!-- ***** Main Banner Area End ***** -->
<section class="services" id="about">
  <br> <br>
```

```
<div class="container">
  <div class="row">
    <div class="col-lg-14">
      <div class="section-heading">
        <button class="btn btn-primary" onclick="lec()">Lecture</button>
        <button class="btn btn-primary" onclick="exam()">Exam</button>
        <h2></h2>
      </div>
    </div>
    <div class="footer">
      <p>Copyright © 2022 Benha Faculty Of Engineering All Rights Reserved.
      <br>Design: <a href="https://www.linkedin.com/in/mahmoud-gouda-409382172/" target="_parent" title="Linkedin page">Mahmoud Gouda</a></p>
    </div>
  </div>
<section class="services" id="lecs">
  <br> <br>
  <div class="container">
    <div class="row">
      <div class="col-lg-14">
        <div class="section-heading">
          <button class="btn btn-primary" onclick="lec()">Lecture</button>
          <button class="btn btn-primary" onclick="exam()">Exam</button>
          <h2></h2>
        </div>
      </div>
    </div>
  </div>
</section>
```

```
</div>

</div>

<datalist id="list1">

    <option value="One"></option>
    <option value="Two"></option>
    <option value="Three"></option>
    <option value="Four"></option>
    <option value="Five"></option>

</datalist>

<form id="login_form" action="" method="post">

    <p class="h6">Year</p>
    <fieldset>
        <input class="form-control" placeholder="The Year ?" list="list1">
<br>

    </fieldset>

    <p class="h6">Course of the exam</p>
    <input class="form-control" type="text" placeholder="The course ?">
    <br>

    <p class="h6">Lecture number:</p>
    <input class="form-control" type="number" placeholder="The lecture number ?" maxlength="2"> <br>

    <button class="btn btn-success" type="submit" action="">Go</button>

</form>

</div>

</div>

<div class="footer">
```

```
<p>Copyright © 2022 Benha Faculty Of Engineering All Rights Reserved.  
<br>Design: <a href="https://www.linkedin.com/in/mahmoud-gouda-409382172/" target="_parent" title="Linkedin page">Mahmoud Gouda</a></p>  
</div>  
</section>  
<section class="services" id="exams">  
  
<br> <br>  
<div class="container">  
  <div class="row">  
    <div class="col-lg-14">  
      <div class="section-heading">  
        <button class="btn btn-primary" onclick="lec()" title="Begin a lecture">Lecture</button>  
        <button class="btn btn-primary" onclick="exam()" title="Create an exam">Exam</button>  
      <h2></h2>  
    </div>  
  </div>  
  <div class="row">  
    <div class="col-lg-14">  
      <h3>List 2</h3>  
      <ul style="list-style-type: none; padding-left: 0;">  
        <li><input type="radio" name="list2"> One</li>  
        <li><input type="radio" name="list2"> Two</li>  
        <li><input type="radio" name="list2"> Three</li>  
        <li><input type="radio" name="list2"> Four</li>  
        <li><input type="radio" name="list2"> Five</li>  
      </ul>  
    </div>  
  </div>  
</div>
```

```
<form id="login_form" action="" method="post">  
    <p class="h6">Year</p>  
    <fieldset>  
        <input class="form-control" placeholder="The Year ?" list="list2">  
<br>  
    </fieldset>  
    <p class="h6">The name of the course of the exam</p>  
    <input class="form-control" type="text" placeholder="The course ?">  
<br>  
    <p class="h6">Exam time</p>  
    <input class="form-control" type="number" placeholder="The Time ?"><br>  
    <p class="h6">Exam number:</p>  
    <input class="form-control" type="number" placeholder="The exam number ?" maxlength="2"> <br>  
    <p class="h6">Questions and answers :</p>  
    <div class="container4">  
        <ul id="demo">  
            <li>  
                <div class="box-2">  
                    <hr color="red" width="70%" size="20" align="center">  
                    <p class="h6">Question</p>  
                    <input class="form-control" type="text" placeholder="Enter the question"><br>  
                    <p class="h6">Answer 1</p>
```

```
<input type="text"><button type="button" class="btn btn-dark"></button>

<p class="h6">Answer 2</p>

<input type="text"><button type="button" class="btn btn-dark"></button>

<p class="h6">Answer 3</p>

<input type="text"><button type="button" class="btn btn-dark"></button>

<p class="h6">Answer 4</p>

<input type="text"><button type="button" class="btn btn-dark"></button><br>

</div>

</li>

</ul>

<hr color="red" width="70%" size="20" align="center">

<button type="button" class="btn btn-primary" title="Add question" onclick="addq()></button>

<button type="button" class="btn btn-primary" title="Delete question" onclick="delq()></button>

<br>

</div><br>

<button class="btn btn-success" type="submit" action="">Submit</button>
```

```
</form>

</div>

</div>

<div class="footer">

    <p>Copyright © 2022 Benha Faculty Of Engineering All Rights Reserved.

        <br>Design: <a href="https://www.linkedin.com/in/mahmoud-gouda-409382172/" target="_parent" title="Linkedin page">Mahmoud Gouda</a></p>

    </div>

</section>

<!-- Scripts -->

<!-- Bootstrap core JavaScript -->

<script src="vendor/jquery/jquery.min.js"></script>

<script src="vendor/bootstrap/js/bootstrap.bundle.min.js"></script>




<script src="assets/js/isotope.min.js"></script>

<script src="assets/js/owl-carousel.js"></script>

<script src="assets/js/lightbox.js"></script>

<script src="assets/js/tabs.js"></script>

<script src="assets/js/video.js"></script>

<script src="assets/js/slick-slider.js"></script>

<script src="assets/js/custom.js"></script>

<script>

    //according to loftblog tut

    function lecture() {

        // body...



```

```
}

</script>

<script>

lecs = document.getElementById("lecs");

exams = document.getElementById("exams");

quest = document.getElementById("quest");

var c=0;

function lec() {

    lecs.style.display="none";

    exams.style.display="none";

    if (lecs.style.display=="none")

        {lecs.style.display = "block";}

    else

        {lecs.style.display = "none";}

}

function exam() {

    lecs.style.display="none";

    exams.style.display="none";

    if (exams.style.display=="none")

        {exams.style.display = "block";}

    else

        {exams.style.display = "none";}

}

}
```

```
function addq() {  
  
    var ul = document.getElementById("demo");  
  
    var li = document.createElement("li");  
  
    li.innerHTML = '<hr color="red" width="70%" size="20"  
align="center"><div class="box-2"><p class="h6">Question</p><input  
class="form-control" type="text" placeholder="Enter the question"><br><p  
class="h6">Answer 1</p><input type="text"><button type="button" class="btn  
btn-dark"></button><p class="h6">Answer 2</p><input type="text"><button  
type="button" class="btn btn-dark"></button><p class="h6">Answer 3</p><input  
type="text"><button type="button" class="btn btn-dark"></button><p class="h6">Answer  
4</p><input type="text"><button type="button" class="btn btn-dark"></button><br>';  
  
    ul.appendChild(li);  
  
}  
  
function delq() {  
  
    var listItems = document.getElementById("demo");  
  
    listItems.removeChild(listItems.lastElementChild);  
  
}  
  
</script>  
</body>  
  
</body>  
</html>
```

(7) Professor personal page

From word personal we can guess what this page includes

This page has the professor personal information registered in the college, the courses the professor teaches and his contact information.

❖ Our brilliant code for this page :

```
<!DOCTYPE html>

<html lang="en-US">

<head>

    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Profile page</title>
    <link href="https://fonts.googleapis.com/css?family=Montserrat:400,700,200" rel="stylesheet">
    <link href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css" rel="stylesheet">
    <link href="css/aos.css?ver=1.1.0" rel="stylesheet">
    <link href="css/bootstrap.min.css?ver=1.1.0" rel="stylesheet">
    <link href="css/main.css?ver=1.1.0" rel="stylesheet">
    <link rel="stylesheet" type="text/css" href="style.css">
    <noscript>
        <style type="text/css">
            [data-aos] {
                opacity: 1 !important;
                transform: translate(0) scale(1) !important;
            }
        </style>
    </noscript>
```

```
</style>
</noscript>
</head>
<body id="top">
<header>
<div class="profile-page sidebar-collapse">
    <nav class="navbar navbar-expand-lg fixed-top navbar-transparent bg-primary" color-on-scroll="400">
        <div class="container">
            <div class="navbar-translate"><a class="navbar-brand" href="/pp" rel="tooltip"></a></div>
            <div class="collapse navbar-collapse justify-content-end" id="navigation">
                <ul class="navbar-nav">
                    <li class="nav-item"><a class="nav-link" href="/proflog" href="#">Home</a></li>
                    <li class="nav-item"><a class="nav-link" href="#about" href="#">About</a></li>
                    <li class="nav-item"><a class="nav-link" href="#experience" href="#">Courses</a></li>
                    <li class="nav-item"><a class="nav-link" href="#contact" href="#">Contact</a></li>
                </ul>
            </div>
        </div>
    </nav>
</div>
```

```
</header>

<div class="page-content">
    <div>
        <div class="profile-page">
            <div class="wrapper">
                <div class="page-header page-header-small" filter-color="green">
                    <div class="page-header-image" data-parallax="true"
                        style="background-image: url('images/project-1.jpg')"></div>
                <div class="container">
                    <div class="content-center">
                        <div class="cc-profile-image"><a href="#"></a></div>
                        <div class="h2 title">[Name from database]</div>
                        <p class="category text-white">[Job from database]</p>
                    </div>
                </div>
                <div class="section">
                    <div class="container">
                        <div class="button-container"><a class="btn btn-default btn-round btn-lg btn-icon" href="#" rel="tooltip" title="Follow me on Facebook"><i class="fa fa-facebook"></i></a><a class="btn btn-default btn-round btn-lg btn-icon" href="#" rel="tooltip" title="Follow me on Twitter"><i class="fa fa-twitter"></i></a><a class="btn btn-default btn-round btn-lg btn-icon" href="#" rel="tooltip" title="Follow me on Google+"><i class="fa fa-google-plus"></i></a><a class="btn btn-default btn-round btn-lg btn-icon" href="#" rel="tooltip" title="Follow me on Instagram"><i class="fa fa-instagram"></i></a></div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
```

```
</div>
</div>
</div>
</div>

<div class="section" id="about">
<div class="container">
<div class="card" data-aos="fade-up" data-aos-offset="10">
<div class="row">
<div class="col-lg-6 col-md-12">
<div class="card-body">
<div class="h4 mt-0 title">About</div>
<p>[Information from database]</p>
</div>
</div>
<div class="col-lg-6 col-md-12">
<div class="card-body">
<div class="h4 mt-0 title">Basic Information</div>
<div class="row">
<div class="col-sm-4"><strong>uppercase</strong></div>
<div class="col-sm-8">[From database]</div>
</div>
<div class="row mt-3">
<div class="col-sm-4"><strong>uppercase</strong></div>
<div class="col-sm-8">[From database]</div>
```

```
</div>

<div class="row mt-3">
    <div class="col-sm-4"><strong>Email:</strong></div>
        <div class="col-sm-8">[From database]</div>
    </div>
    <div class="row mt-3">
        <div class="col-sm-4"><strong>Phone:</strong></div>
        <div class="col-sm-8">[From database]</div>
    </div>
    </div>
    </div>
    </div>
</div>

<div class="section" id="experience">
    <div class="container cc-experience">
        <div class="h4 text-center mb-4 title">Courses</div>
        <div class="card">
            <div class="row">
                <div class="col-md-3 bg-primary" data-aos="fade-right" data-aos-offset="50" data-aos-duration="500">
                    <div class="card-body cc-experience-header">
                        <div class="h5">[Year of the course from database]</div>

```

```
</div>

</div>

<div class="col-md-9" data-aos="fade-left" data-aos-offset="50"
data-aos-duration="500">

    <div class="card-body">
        <div class="h5">[Name of the course1 from database]</div>
        <p>[Description of the course from database (added by
admin)]</p>
    </div>
</div>

</div>

</div>

</div>

<div class="card">
    <div class="row">
        <div class="col-md-3 bg-primary" data-aos="fade-right" data-aos-
offset="50" data-aos-duration="500">
            <div class="card-body cc-experience-header">
                <div class="h5">[Year of the course from database]</div>
            </div>
        </div>
        <div class="col-md-9" data-aos="fade-left" data-aos-offset="50"
data-aos-duration="500">
            <div class="card-body">
                <div class="h5">[Name of the course2 from database]</div>
                <p>[Description of the course from database (added by
admin)]</p>
            </div>
        </div>
    </div>
</div>
```

```
</div>
</div>
</div>
</div>
</div>
<div class="card">
<div class="row">
    <div class="col-md-3 bg-primary" data-aos="fade-right" data-aos-offset="50" data-aos-duration="500">
        <div class="card-body cc-experience-header">
            <div class="h5">[Year of the course from database]</div>
        </div>
    </div>
    <div class="col-md-9" data-aos="fade-left" data-aos-offset="50" data-aos-duration="500">
        <div class="card-body">
            <div class="h5">[Name of the course3 from database]</div>
            <p>[Description of the course from database (added by admin)]</p>
        </div>
    </div>
</div>
</div>
</div>
</div>
</div>
<div class="section" id="contact">
```

```
<div      class="cc-contact-information"      style="background-image:  
url('images/project-3.jpg')">  
  
<div class="container">  
  
<div class="cc-contact">  
  
<div class="row">  
  
<div class="col-md-9">  
  
<div class="card mb-0" data-aos="zoom-in">  
  
<div class="h4 text-center title">Contact Me</div>  
  
<div class="row">  
  
<div class="col-md-6">  
  
<div class="card-body">  
  
<form  
action="https://formspree.io/your@email.com" method="POST">  
  
<div class="p pb-3"><strong>Feel free to contact  
me </strong></div>  
  
<div class="row mb-3">  
  
<div class="col">  
  
<div      class="input-group"><span  
class="input-group-addon"><i class="fa fa-user-circle"></i></span>  
  
<input class="form-control , use-keyboard-  
input" type="text" name="name" placeholder="Name" required="required"/>  
  
</div>  
  
</div>  
  
</div>  
  
<div class="row mb-3">  
  
<div class="col">
```

```
<div class="input-group"><span
class="input-group-addon"><i class="fa fa-file-text"></i></span>
<input class="form-control , use-keyboard-
input" type="text" name="Subject" placeholder="Subject" required="required"/>
</div>
</div>
</div>
<div class="row mb-3">
<div class="col">
<div class="input-group"><span
class="input-group-addon"><i class="fa fa-envelope"></i></span>
<input class="form-control , use-keyboard-
input" type="email" name="_replyto" placeholder="E-mail" required="required"/>
</div>
</div>
</div>
<div class="row mb-3">
<div class="col">
<div class="form-group">
<textarea class="form-control , use-
keyboard-input" name="message" placeholder="Your Message"
required="required"></textarea>
</div>
</div>
</div>
<div class="row">
<div class="col">
```

```
        <button          class="btn"      btn-primary">
type="submit">Send</button>

        </div>
    </div>
</form>
</div>
</div>
<div class="col-md-6">
    <div class="card-body">
        <p class="mb-0"><strong>Phone</strong></p>
        <p class="pb-2">[From database]</p>
        <p class="mb-0"><strong>Email</strong></p>
        <p>[From database]</p>
    </div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div></div>
</div>
<footer class="footer">
```

```
<div class="container text-center"><a class="cc-facebook btn btn-link" href="#"><i class="fa fa-facebook fa-2x " aria-hidden="true"></i></a><a class="cc-twitter btn btn-link " href="#"><i class="fa fa-twitter fa-2x " aria-hidden="true"></i></a><a class="cc-google-plus btn btn-link" href="#"><i class="fa fa-google-plus fa-2x" aria-hidden="true"></i></a><a class="cc-instagram btn btn-link" href="#"><i class="fa fa-instagram fa-2x " aria-hidden="true"></i></a></div>

<div class="h4 title text-center">Benha Faculty of Engineering</div>

<div class="text-center text-muted">
    <p>&copy; COPYRIGHT © 2022 BENHA FACULTY OF ENGINEERING  

    ALL RIGHTS RESERVED.<br>Design - <a class="credit" href="https://www.linkedin.com/in/mahmoud-gouda-409382172/" target="_blank">Mahmoud Gouda</a></p>
</div>
</footer>

<script src="js/core/jquery.3.2.1.min.js?ver=1.1.0"></script>
<script src="js/core/popper.min.js?ver=1.1.0"></script>
<script src="js/core/bootstrap.min.js?ver=1.1.0"></script>
<script src="js/now-ui-kit.js?ver=1.1.0"></script>
<script src="js/aos.js?ver=1.1.0"></script>
<script src="scripts/main.js?ver=1.1.0"></script>
<script src="keyboard.js"></script>
</body>
</html>
```

(8) The lecture live page for the student

We can say this page have the controls which is available for the student during the lecture:

- *Raise hand*: This allows the student to raise his hand to ask for the professor permission to take about something.
- *Send a question*: If we have shy students, and it may exist, we allow them to ask there question by writing it without speaking or raising hands.
- *Vote*: Sometimes, the professor need a vote from the students to something related to the learning process, so this control allow the students to vote in what is shown.
- *Quiz*: If the professor needs to make a quiz in the lecture, the student can show the question from this control and answer it.

Besides the controls, this page has information about the current lecture, like the name of the course and course number.

❖ Our brilliant code for this page :

```
<!DOCTYPE html>

<html>
<head>

<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"></script>
<link rel="stylesheet"
      href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.css"
      integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cYiJTQUOhcWr7x9JvoRxT2M
      Zw1T" crossorigin="anonymous">
<link rel="stylesheet" type="text/css" href="try.css">
<link rel="stylesheet" type="text/css" href="try.js">
```

```
<link rel="stylesheet" type="text/css" href="style.css">
<title>Home Page</title>
</head>
<body>
<h2 style="background:#302ea3; color:white; padding:10px 20px;">
    Lecture one : Antenna Analysis and design
</h2>
<div id="stream"><iframe src="https://www.youtube.com/embed/wu8pF5B-rY8" width="888" height="499" allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe></div>

<div class="chat-popup" id="myForm">
    <form class="form-container">
        <h2>Type your question</h2>
        <textarea placeholder="Type here.." class="use-keyboard-input" name="msg" required></textarea>
        <button type="submit" class="btn">Send</button>
        <button type="button" class="btn" onclick="closeForm()">Close</button>
    </form>
</div>
<div class="chat-popup" id="quiz">
    <form class="form-container">
        <h2>The QUESTION: ....?</h2>
        <textarea placeholder="Type here.." class="use-keyboard-input" name="msg" required></textarea>
```

```
<button type="submit" class="btn" onclick="closeForm1()">SUMBIT THE  
ANSWER</button>  
</form>  
</div>  
<div class="popup-vote" id="popup-vote">  
    <p class="h6">Question : [Get from database]</p>  
    <input type="radio" name="theSame"><p>choice1</p>  
    <input type="radio" name="theSame"><p>choice2</p>  
    <input type="radio" name="theSame"><p>choice3</p>  
    <input type="radio" name="theSame"><p>choice4</p>  
    <button type="button" onclick="closepopup()">Send</button>  
</div>  
<footer>  
  
<ul>  
    <li> <button class="btn btn-dark" id="btn"></button></li>  
    <li> <button onclick="getLocalStream()" class="btn btn-dark"></button></li>  
    <li><button onclick="chat()" class="btn btn-dark"></button></li>  
    <li> <button class="btn btn-dark" onclick="openpopup()" title="Vote">
```

```

</button>
</li>
<li>
    <button onclick="quiz()" class="btn btn-dark"></button>
</li>
</ul>
</footer>

<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script type="text/javascript">
// jQuery Document
$(document).ready(function () {});
</script>

<script >
function getLocalStream() {
    navigator.mediaDevices.getUserMedia({video: false, audio: true}).then(
stream => {
    window.localStream = stream; // A
    window.localAudio.srcObject = stream; // B
    window.localAudio.autoplay = true; // C
}).catch( err => {
```

```
        console.log("u got an error:" + err)
    });
}

const btn = document.getElementById('btn');
btn.addEventListener('click', function onclick() {
    btn.classList.toggle("onclick_style")
});

</script>

<script>
function chat() {
    document.getElementById("myForm").style.display = "block";
    document.getElementById("quiz").style.display = "none";
}

function closeForm() {
    document.getElementById("myForm").style.display = "none";
}

function quiz() {
    document.getElementById("quiz").style.display = "block";
    document.getElementById("myForm").style.display = "none";
}

function closeForm1() {
```

```
document.getElementById("quiz").style.display = "none";  
}  
  
let mypop = document.getElementById("popup-vote")  
  
function openpopup()  
{  
    mypop.classList.add("open-popup");  
}  
  
function closepopup()  
{  
    mypop.classList.remove("open-popup");  
}  
  
</script>  
  
<script src="keyboard.js"></script>  
  
</body>  
</html>
```

(9) The lecture live page for the Professor

Like the student live page, this page has controls as well but this time for the professor during the lecture:

- *Attendees*: this control shows the professor who is attending the lecture at the moment.
- *Questions*: we said previously that the students are allowed to send questions during the lecture. Here in this control, these questions are shown for the professor and a button to remove it after answering the question.
- *Vote*: here, the professor send the question and the answers available for the vote he needs to make.

- *Quiz*: If the professor wants to make a quiz during the lecture, he writes it here and its grade.

Besides the controls, this page has also information about the current lecture, like the name of the course and course number.

❖ Our brilliant code for this page :

```
<!DOCTYPE html>

<html>

<head>

    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"></script>
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.css"
integrity="sha384-ggOYR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cYiJTQUOhcWr7x9JvoRxT2M
Zw1T" crossorigin="anonymous">
    <link rel="stylesheet" type="text/css" href="try.css">
    <link rel="stylesheet" type="text/css" href="try.js">
    <link rel="stylesheet" type="text/css" href="style.css">
    <title>Home Page</title>
</head>

<body>

    <h2 style="background:#302ea3; color:white; padding:10px 20px;">
```

Lecture one : Antenna Analysis and design

```
</h2>

<div id="stream"><iframe src="https://docs.google.com/presentation/d/17aFAYmf8ZUHc5y45znfZBh4o3tqJcc3K/edit?usp=sharing&ouid=101575391344746801298&rtpof=true&sd=true"></iframe></div>

<div class="chat-popup" id="myForm">
  <form class="form-container">
    <h2>Questions from students</h2>
    <div id="student1" >
      <P><h5>Question from (Get the name from the database) :</h5>
      [The question from the database] <button type="button" class="img-center" onclick="endq1()"></button>
      </P>
    </div>
    <div class="line-5"></div>
    <div id="student2" >
      <P><h5>Question from (Get the name from the database) :</h5>
      [The question from the database] <button type="button" class="img-center" onclick="endq2()"></button>
      </P>
    </div>
    <button type="button" class="btn" cancel" onclick="closeForm()">Close</button>
```

```
</form>

</div>

<div class="chat-popup" id="quiz">

  <form class="form-container">

    <h2>The QUESTION: ....?</h2>

    <textarea placeholder="Type here.." class="use-keyboard-input" name="msg" required></textarea>

    <h2>The grade: ....?</h2>

    <textarea placeholder="Type here.." class="use-keyboard-input" name="msg" required></textarea>

    <button type="button" class="btn" onclick="sent_quiz()">Send the quiz</button>

    <button type="button" class="btn" onclick="closeForm1()">Close</button>

  </form>

</div>

<div class="chat-popup" id="attend">

  <form class="form-container">

    <h2>Check who is attending ...</h2>

    <h3>Attendees</h3>

    <p>Get the names from the database ....</p>

    <p>

      Ahmed

      <button class="img-rounded" type="button"></button>

      <button class="img-rounded" type="button"></button>

    </p>

  </form>

</div>
```

```
</p>

<h3>Absentees</h3>

<p>Get the names from the database ....</p>

<button type="button" class="btn" cancel" onclick="closeForm2()">Close</button>

</form>

</div>

<div class="chat-popup" id="voice">

<form class="form-container">

    <h2>Allow students to speek</h2><button class="img-rounded" type="button"></button>

    <p>Get the names from the database ....</p>

    <!-- As a example -->

    <h4>Mohamed Ahmed <button id="btn" type="button"></button></h4>

    <button type="button" class="btn" cancel" onclick="closeForm3()">Close</button>

</form>

</div>

<div class="chat-popup" id="quiz_answer">

<form class="form-container">

    <h2>Answers to the quiz</h2>

    <!-- As example -->

    <p>Ahmed [name from the database]</p>
```

```
<p>
    The answer is : [The answer from the database]
    <button id="bonus" type="button"></button>
    <button id="zero" type="button"></button>
</p>
<div class="line-5"></div>
<p>Mohamed [name from the database]</p>
<p>The answer is : [The answer from the database]
    <button id="bonus" type="button"></button>
    <button id="zero" type="button"></button>
</p>

<button type="button" class="btn cancel" onclick="closequiz()>Close</button>
</form>
</div>
<div class="popup-vote" id="popup-vote">
    <div>
        <p class="h6">Question</p>
        <input class="form-control" type="text" placeholder="Enter the question"><br>
        <ul class="list-group" id="the_q">
```

```
<li>
    <p>Answer</p>
    <input type="text">
</li>
<li>
    <p>Answer</p>
    <input type="text">
</li>
</ul>
</div>
<div>
    <button type="button" onclick="add_answer()">Add</button>
    <button type="button" onclick="del_answer()">Delete</button>
    <button type="button" onclick="closepopup()">Send</button>
    <button type="button" onclick="canclepop()">Cancel</button>
</div>
</div>
<footer>

<ul>
    <li> <button class="btn btn-dark"></button></li>
    <li> <button onclick="voice()" class="btn btn-dark"></button></li>
```

```
<li><button onclick="chat()" class="btn btn-dark"></button></li>

<li >
    <button class="btn btn-dark" onclick="openpopup()" title="Vote">
        
    </button>
</li>
<li>
    <button onclick="quiz()" class="btn btn-dark"></button>
</li>
</ul>
</footer>

<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script type="text/javascript">
    // jQuery Document
    $(document).ready(function () {});
</script>

<script>
    function voice() {
        document.getElementById("voice").style.display = "block";
    }
</script>
```

```
document.getElementById("myForm").style.display = "none";
document.getElementById("quiz").style.display = "none";
document.getElementById("attend").style.display = "none";
}

const btn = document.getElementById('btn');
btn.addEventListener('click', function onclick() {
    if(btn.style.color = "white")
        document.getElementById("btn").style.color = "red";
    else
        document.getElementById("btn").style.color = "green";
});
```

</script>

```
<script>
function chat() {
    document.getElementById("myForm").style.display = "block";
    document.getElementById("quiz").style.display = "none";
    document.getElementById("voice").style.display = "none";
    document.getElementById("attend").style.display = "none";
}
```

```
function closeForm() {
    document.getElementById("myForm").style.display = "none";
}
```

```
function quiz() {  
    document.getElementById("myForm").style.display = "none";  
    document.getElementById("quiz").style.display = "block";  
    document.getElementById("voice").style.display = "none";  
    document.getElementById("attend").style.display = "none";  
  
}  
  
function attend() {  
    document.getElementById("myForm").style.display = "none";  
    document.getElementById("quiz").style.display = "none";  
    document.getElementById("voice").style.display = "none";  
    document.getElementById("attend").style.display = "block";  
  
}  
  
function closequiz() {  
    document.getElementById("myForm").style.display = "none";  
    document.getElementById("quiz").style.display = "none";  
    document.getElementById("voice").style.display = "none";  
    document.getElementById("attend").style.display = "none";  
    document.getElementById("quiz_answer").style.display = "none";  
  
}  
  
function closeForm1() {  
    document.getElementById("myForm").style.display = "none";  
    document.getElementById("quiz").style.display = "none";  
    document.getElementById("voice").style.display = "none";  
    document.getElementById("attend").style.display = "none";  
}
```

```
}

function sent_quiz() {
    document.getElementById("myForm").style.display = "none";
    document.getElementById("quiz").style.display = "none";
    document.getElementById("voice").style.display = "none";
    document.getElementById("attend").style.display = "none";
    document.getElementById("quiz_answer").style.display = "block";
}

function closeForm2() {
    document.getElementById("attend").style.display = "none";
}

function closeForm3() {
    document.getElementById("voice").style.display = "none";
}

function endq1() {
    document.getElementById("student1").style.display = "none";
}

function endq2() {
    document.getElementById("student2").style.display = "none";
}

let mypop = document.getElementById("popup-vote")

function openpopup()
{
    mypop.classList.add("open-popup");
}
```

```
}

function closepopup()
{
    mypop.classList.remove("open-popup");
}

function canclepop()
{
    mypop.classList.remove("open-popup");
}

function add_answer() {
    var ul = document.getElementById("the_q");
    var li = document.createElement("li");
    li.innerHTML = '<p class="h6">Answer</p><input type="text"><br>';
    ul.appendChild(li);
}

function del_answer() {
    var listItems = document.getElementById("the_q");
    listItems.removeChild(listItems.lastElementChild);
}

</script>
<script src="keyboard.js"></script>
</body>
</html>
```

(10) Main page

Above, we talked about all pages that are available for students, professors and admins. This leads us to question: what is available for guests?

Well, here we are going to answer this question. In our design we made a page that is available for everyone to know about the college.

So, we mainly filled this page with information about the courses are taught in our college for all years.

This supplies any guest with what he need to know about the college, so who is interested can apply for this college later.

Also, we added a field for guests to ask questions or leave a comment. These are all the guests can know or do.

❖ Our brilliant code for this page :

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="TemplateMo">
    <link href="https://fonts.googleapis.com/css?family=Poppins:100,200,300,400,500,600,700,800,900" rel="stylesheet">
```

```
<title>Engineering</title>

<!-- Bootstrap core CSS --&gt;
&lt;link href="vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet"&gt;
<!-- Additional CSS Files --&gt;
&lt;link rel="stylesheet" href="assets/css/fontawesome.css"&gt;
&lt;link rel="stylesheet" href="assets/css/templatemo-edu-meeting.css"&gt;
&lt;link rel="stylesheet" href="assets/css/owl.css"&gt;
&lt;link rel="stylesheet" href="assets/css/lightbox.css"&gt;
&lt;link rel="stylesheet" type="text/css" href="login.css"&gt;
&lt;script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"&gt;&lt;/script&gt;
&lt;link rel="stylesheet" type="text/css" href="style.css"&gt;
&lt;/head&gt;

&lt;body&gt;

<!-- Sub Header --&gt;
&lt;div class="sub-header"&gt;
&lt;div class="container"&gt;
&lt;div class="row"&gt;
&lt;div class="col-lg-8 col-sm-8"&gt;
&lt;div class="left-content"&gt;
&lt;p&gt;Welcome to &lt;em&gt;Benha University&lt;/em&gt;&lt;/p&gt;
&lt;/div&gt;</pre>
```

```
</div>
</div>
</div>
</div>

<!-- ***** Header Area Start ***** -->
<header class="header-area header-sticky">
<div class="container">
<div class="row">
<div class="col-12">

<nav class="main-nav">
<!-- ***** Logo Start ***** -->
<br> <br>
<a href="index.html" class="logo">
    Benha Faculty of Engineering
</a>
<!-- ***** Logo End ***** -->
<!-- ***** Menu Start ***** -->
<ul class="nav">
    <li class="scroll-to-section"><a href="#top" href="#" class="active">Home</a></li>
    <li><a href="#about">About</a></li>
    <li class="scroll-to-section"><a href="#contact" href="#">Contact Us</a></li>
    <li>
```

```
<a href="/login"><button class="btn btn-primary">Log
in</button></a>

</li>
</ul>
<a class='menu-trigger'>
  <span>Menu</span>
</a>
<!-- ***** Menu End ***** -->
</nav>
</div>
</div>
</div>
</header>
<!-- ***** Header Area End ***** -->
<!-- Model -->
<div class="modal fade" id="myModal">
  <div class="modal-dialog modal-dialog-centered" >
    <div class="modal-content">
      <!-- Header -->
      <div class="modal-header">
        <h4 class="modal-title">LOGIN</h4>
        <button type="button" class="close" data-dismiss="modal">x</button>
      </div>
      <!--body -->
      <div class="modal-body">
```

```
<form>

    <p id="log">LOGIN AS</p>
    <input class="field_class" list="list1" placeholder="Who are you?">
    <datalist id="list1">
        <option value="STUDENT"></option>
        <option value="PROFESSOR"></option>
        <option value="ADMIN"></option>
    </datalist>

    <br>
    <div class="form_div">
        <p id="no1">ID:</p>
        <input class="field_class" name="login_txt" type="text"
placeholder="Enter your ID" maxlength="20">
        <p id="no2">PASSWORD:</p>
        <input id="pass" class="field_class" name="password_txt"
type="password" placeholder="Enter the password" maxlength="20">
        <button class="submit_class" type="submit"
form="login_form">Submit</button>
    </div>
</form>

</div>
<!-- Footer --&gt;
&lt;div class="modal-footer"&gt;</pre>
```

```
<button type="button" class="btn btn-danger" data-
dismiss="modal">Close</button>

</div>
</div>
</div>
</div>

<!-- Model -->

<!-- ***** Main Banner Area Start ***** -->

<section class="section main-banner" id="top" data-section="section1">
  <video autoplay muted loop id="bg-video">
    <source src="assets/images/vv.mp4" type="video/mp4" />
  </video>

  <div class="video-overlay header-text">
    <div class="container">
      <div class="row">
        <div class="col-lg-12">
          <div class="caption">
            <h6>Hello Students</h6>
            <h2>Welcome to Benha Faculty Of Engineering</h2>
            <p>Our target is to keep our students in touch with modern technologies related to electrical engineering.</p>
          </div>
        </div>
      </div>
    </div>
  </div>
```

```
</div>

<br> <br> <br> <br> <br> <br> <br> <br>

</section>

<!-- ***** Main Banner Area End ***** -->
```

```
<section class="services" id="about">
```

```
    <br> <br>
```

```
    <div class="container">
```

```
        <div class="row">
```

```
            <div class="col-lg-14">
```

```
                <div class="section-heading">
```

```
                    <h2>About our Courses</h2>
```

```
                    <p> Here, you'll find 5 years of learning, we don't have two semester  
in one years it's only one.</p>
```

```
                    <p>Below, you will find all courses you study in each year.</p>
```

```
                </div>
```

```
            </div>
```

```
<br>
```

```
    <br> <br> <br> <br> <br> <br>
```

```
    <div class="col-lg-12">
```

```
        <div class="owl-service-item owl-carousel">
```

```
            <div class="item">
```

```
<div class="icon">  
      
</div>  
  
<div class="down-content">  
    <h4>First Year</h4>  
    <ul>  
        <li>Math 1,2</li>  
        <li>Physics</li>  
        <li>Ethics</li>  
        <li>Information and computer technology</li>  
        <li>Python</li>  
    </ul>  
</div>  
</div>
```

```
<div class="item">  
    <div class="icon">  
          
    </div>  
    <div class="down-content">  
        <h4>Second Year</h4>  
        <ul>  
            <li>Math 3,4</li>  
            <li>Modern Physics</li>  
            <li>C++</li>  
        </ul>  
    </div>  
</div>
```

```
<li>Circuit Analysis</li>
<li>Login 1,2</li>
<li>Environment safety</li>
</ul>
</div>
</div>

<div class="item">
<div class="icon">

</div>
<div class="down-content">
<h4>Third Year</h4>
<ul>
<li>Random Process</li>
<li>Computer Architecture</li>
<li>Matlab</li>
<li>Field</li>
<li>Circuit design</li>
<li>Electric measurements</li>
</ul>
</div>
</div>

<div class="item">
```

```
<div class="icon">  
      
</div>  
  
<div class="down-content">  
    <h4>Fourth Year</h4>  
    <ul>  
        <li>Network</li>  
        <li>Transmition line</li>  
        <li>Analog communication</li>  
        <li>Information theory</li>  
        <li>Embedded systems</li>  
        <li>Computer organization</li>  
    </ul>  
</div>  
</div>  
  
<div class="item">  
    <div class="icon">  
          
</div>  
    <div class="down-content">  
        <h4>Fifth Year</h4>  
        <ul>  
            <li>Digital communication</li>  
            <li>Digital signal processing</li>  
        </ul>  
    </div>  
</div>
```

```
<li>Anvanced Network</li>
<li>Operating systems</li>
<li>Image processing</li>
<li>Network security</li>
</ul>
</div>
</div>

</div>
</div>
</div>
</div>
<br>
<br> <br> <br> <br>
</section>
<br><br><br><br><br><br> <br> <br> <br> <br> <br> <br> <br>
<br> <br> <br> <br>
<section class="our-facts">
<div class="container">
<div class="row">
<div class="col-lg-6">
<div class="row">
<div class="col-lg-12">
<h2>A Few Facts About Our College</h2>
```

```
</div>

<div class="col-lg-6">
    <div class="row">
        <div class="col-12">
            <div class="count-area-content percentage bg-dark">
                <div class="count-digit">94</div>
                <div class="count-title">Succesed Students</div>
            </div>
        </div>
        <div class="col-12">
            <div class="count-area-content bg-dark">
                <div class="count-digit">126</div>
                <div class="count-title">Current Teachers</div>
            </div>
        </div>
    </div>
</div>

<div class="col-lg-6">
    <div class="row">
        <div class="col-12">
            <div class="count-area-content new-students bg-dark">
                <div class="count-digit">2345</div>
                <div class="count-title">New Students</div>
            </div>
        </div>
    </div>
```

```
<div class="col-12">  
    <div class="count-area-content bg-dark">  
        <div class="count-digit">32</div>  
        <div class="count-title">Awards</div>  
    </div>  
    </div>  
    </div>  
    </div>  
    </div>  
    </div>  
    <div class="col-lg-6 align-self-center">  
        <div class="video">  
            <a href="https://www.youtube.com/watch?v=HndV87XpkWg" target="_blank"></a>  
        </div>  
        </div>  
    </div>  
    </div>  
</section>  
  
<section class="contact-us" id="contact">  
    <div class="container">  
        <div class="row">  
            <div class="col-lg-9 align-self-center">  
                <div class="row">
```

```
<div class="col-lg-12">
    <form id="contact" action="" method="post">
        <div class="row">
            <div class="col-lg-12">
                <h2>Let's get in touch</h2>
            </div>
            <div class="col-lg-4">
                <fieldset>
                    <input name="name" type="text" id="name" class="use-keyboard-input" placeholder="YOURNAME...*" required="">
                </fieldset>
            </div>
            <div class="col-lg-4">
                <fieldset>
                    <input name="email" type="text" id="email" class="use-keyboard-input" pattern="[^ @]*@[^ @]*" placeholder="YOUR EMAIL..." required="">
                </fieldset>
            </div>
            <div class="col-lg-4">
                <fieldset>
                    <input name="subject" type="text" class="use-keyboard-input" id="subject" placeholder="SUBJECT...*" required="">
                </fieldset>
            </div>
        <div class="col-lg-12">
```

```
<fieldset>

    <textarea name="message" class="use-keyboard-input"
type="text" class="form-control" id="message" placeholder="YOUR
MESSAGE..." required=""></textarea>

</fieldset>

</div>

<div class="col-lg-12">

    <fieldset>

        <button type="submit" id="form-submit"
class="button">SEND MESSAGE NOW</button>

    </fieldset>

</div>

</div>

</form>

</div>

</div>

</div>

<div class="col-lg-3">

    <div class="right-info">

        <ul>

            <li>

                <h6>Phone Number</h6>

                <span>010-020-0340</span>

            </li>

            <li>

                <h6>Email Address</h6>

            </li>

        </ul>

    </div>

</div>
```

```
<span>Example@beng.bu.edu.eg</span>
</li>
<li>
    <h6>Address</h6>
    <span>The stadium st., Benha</span>
</li>
</ul>
</div>
</div>
</div>
<div class="footer">
    <p>Copyright © 2022 Benha Faculty Of Engineering All Rights Reserved.
        <br>Design: <a href="https://www.linkedin.com/in/mahmoud-gouda-409382172/" target="_parent" title="free css templates">Mahmoud Gouda</a></p>
</div>
</section>

<!-- Scripts -->
<!-- Bootstrap core JavaScript -->
<script src="vendor/jquery/jquery.min.js"></script>
<script src="vendor/bootstrap/js/bootstrap.bundle.min.js"></script>

<script src="assets/js/isotope.min.js"></script>
<script src="assets/js/owl-carousel.js"></script>
```

```
<script src="assets/js/lightbox.js"></script>
<script src="assets/js/tabs.js"></script>
<script src="assets/js/video.js"></script>
<script src="assets/js/slick-slider.js"></script>
<script src="assets/js/custom.js"></script>
<script>
    //according to loftblog tut
    $('.nav li:first').addClass('active');

var showSection = function showSection(section, isAnimate) {
    var
        direction = section.replace(/\#/,""),
        reqSection = $('.section').filter('[data-section="' + direction + '"]'),
        reqSectionPos = reqSection.offset().top - 0;

    if (isAnimate) {
        $('body, html').animate({
            scrollTop: reqSectionPos },
            800);
    } else {
        $('body, html').scrollTop(reqSectionPos);
    }
};


```

```
var checkSection = function checkSection() {
    $('.section').each(function () {
        var
            $this = $(this),
            topEdge = $this.offset().top - 80,
            bottomEdge = topEdge + $this.height(),
            wScroll = $(window).scrollTop();
        if (topEdge < wScroll && bottomEdge > wScroll) {
            var
                currentId = $this.data('section'),
                reqLink = $('a').filter('[href*=\#\'' + currentId + ']');
            reqLink.closest('li').addClass('active').
                siblings().removeClass('active');
        }
    });
};

$('.main-menu, .responsive-menu, .scroll-to-section').on('click', 'a', function (e) {
    e.preventDefault();
    showSection($(this).attr('href'), true);
});

$(window).scroll(function () {
    checkSection();
});
```

```
</script>  
  
<script src="keyboard.js"></script>  
  
</body>  
  
</body>  
  
</html>
```

4.4- The exam part

You may wonder: where is the page that shows the exam for students?

As we know we made the teaching process in a local network, but what if a problem happened in this local network during the exam? This made us think of it and we suggested to make the exam as a python GUI for the student in the raspberry pi, so if a problem happened in the network, the student could complete the exam without any problem.

4.5- How all pages are connected to the server?

While moving between these webpages, we will find out that we need two methods to handle data from our database.

First of all, we have pages that needs to get or post data from database that is not urgent to be used in the same moment. This type deals with PHP language which is the best choice for this job. As it is simple to write and allows us to handle needed data from our database easily. Also, we have pages that needs data urgently which we call a real-time type. This part is related to the pages designed for lectures as when something is happening right now, we need any data transferred is shown to all members as long as it is written and sent. We have researched for a solution for this type and we found out that Websocket is the best way for this communication. We will take about it in a separated chapter to explain it in details.

4.6 Conclusion

We used HTML to design pages that are available for everyone with their own authorization in order to benefit each kind of people with what they need. We used Webscoket and PHP in order to communicate with the server and each one is used in its own application.

5. Chapter Five: WebSocket

5.1 Introduction:

Smart Class-room depend mainly on the concept of real-time communication between students and their professors, this brings many challenges and obstacles. In this chapter we will go through how we used many web standard to achieve full-duplex bidirectional communication that is used in the smart classroom, we will begin with a broad discussion of different ways to achieve this functionality and the advantages and disadvantages of each approach.

After that we will move to different internet protocols like XMPP that are built upon WebSocket and see how they are implemented and how different applications tweak them for their need. Then we will discuss how we implemented a simple version of XMPP [RFC 6120] [1] standard core to meet our requirement. Finally we will go through implementation of this protocol using [php ratchet library] [2] and discuss various software engineering aspects.

Bidirectional Stateful Communication

The internet began with Web 1.0 which refers to the first stage of the World Wide Web evolution. Earlier, there were only a few content creators in Web 1.0 with a huge majority of users who are consumers of content. Personal web pages were common, consisting mainly of static pages hosted on ISP-run web servers, or free web hosting services.

This approach continues to rule even when Web 2.0 came out we still have high-end servers which contain multiple datasets and clients who use these services to obtain the information they need for example when you want to visit example.php the process goes as follows

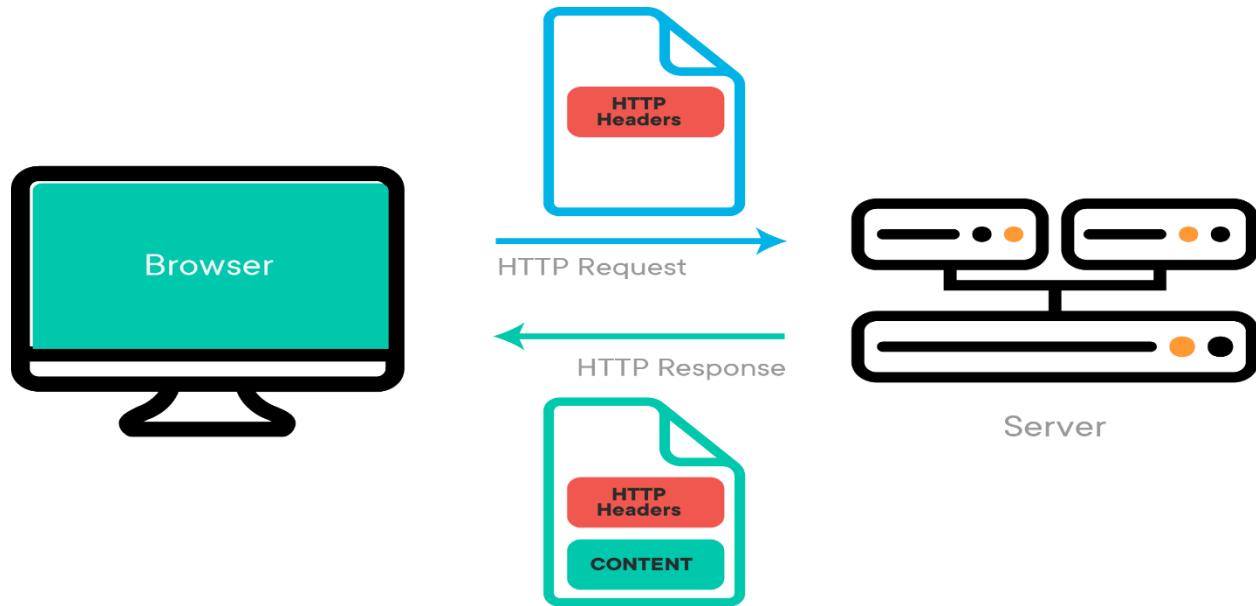


Figure 5-1. HTTP Protocol

This is a stateless one directional connection between you and the server, you send the request and the server responds. There is no open continuous direct connection between you and the server.

Till now everything looks great but how about when we need to keep an active connection with the server where data can flow in both directions in same time in real-time like messengers application. The request-response pipeline that we have discussed so far is not the right solution to this dilemma, Next, we will see possible answers to this question.

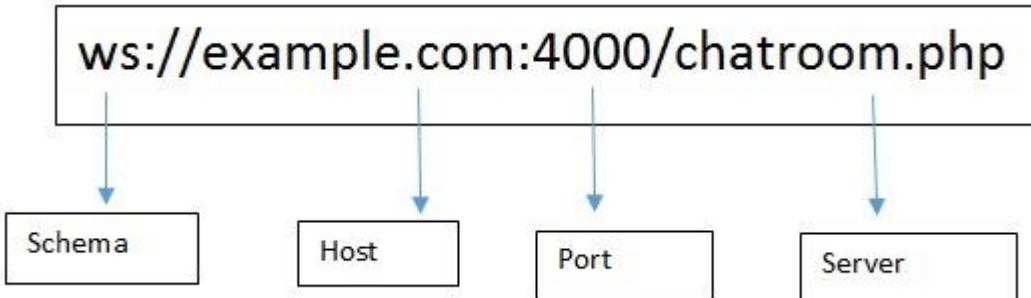
5.2 WebSocket:

The WebSocket protocol, described in the specification RFC 6455, provides a way to exchange data between browser and server via a persistent connection. The data can be passed in both directions as “packets”, without breaking the connection and the need for additional HTTP requests.

Great use cases for WebSockets includes:

- Chat Applications
- Multiplayer Games

- Collaborative Editing
- Social Feeds
- Location-based Applications



Server-Sent Events

The Server-Sent Events specification describes a built-in class `EventSource`, that keeps a connection with the server and allows to receive events from it.

Similar to WebSocket, the connection is persistent.

But there are several important differences:

Table 5-1. Comparison between WebSocket and EventSource

WebSocket	EventSource
Bi-directional: both client and server can exchange messages	One-directional: the only server sends data
Binary and text data	Only text
WebSocket protocol	Regular HTTP

5.3 Long polling

Long polling is the simplest way of having a persistent connection with the server, that doesn't use any specific protocol like WebSocket or Server Side Events.

Being very easy to implement, it's also good enough in a lot of cases.

Regular Polling The simplest way to get new information from the server is periodic polling. That is, regular requests to the server: “Hello, I’m here, do you have any information for me?”. For example, once every 10 seconds.

In response, the server first takes a notice to itself that the client is online, and second – sends a packet of messages it got till that moment.

That works, but there are downsides:

Messages are passed with a delay up to 10 seconds (between requests). Even if there are no messages, the server is bombed with requests every 10 seconds, even if the user switched somewhere else or is asleep. That’s quite a load to handle, speaking performance-wise. So, if we’re talking about a very small service, the approach may be viable, but generally, it needs an improvement.

Long polling So-called “long polling” is a much better way to poll the server.

It’s also very easy to implement and delivers messages without delays.

The flow:

A request is sent to the server. The server doesn’t close the connection until it has a message to send. When a message appears – the server responds to the request with it. The browser makes a new request immediately. The situation when the browser sent a request and has a pending connection with the server is standard for this method. Only when a message is delivered, the connection is reestablished. If the connection is lost, because of, say, a network error, the browser immediately sends a new request.

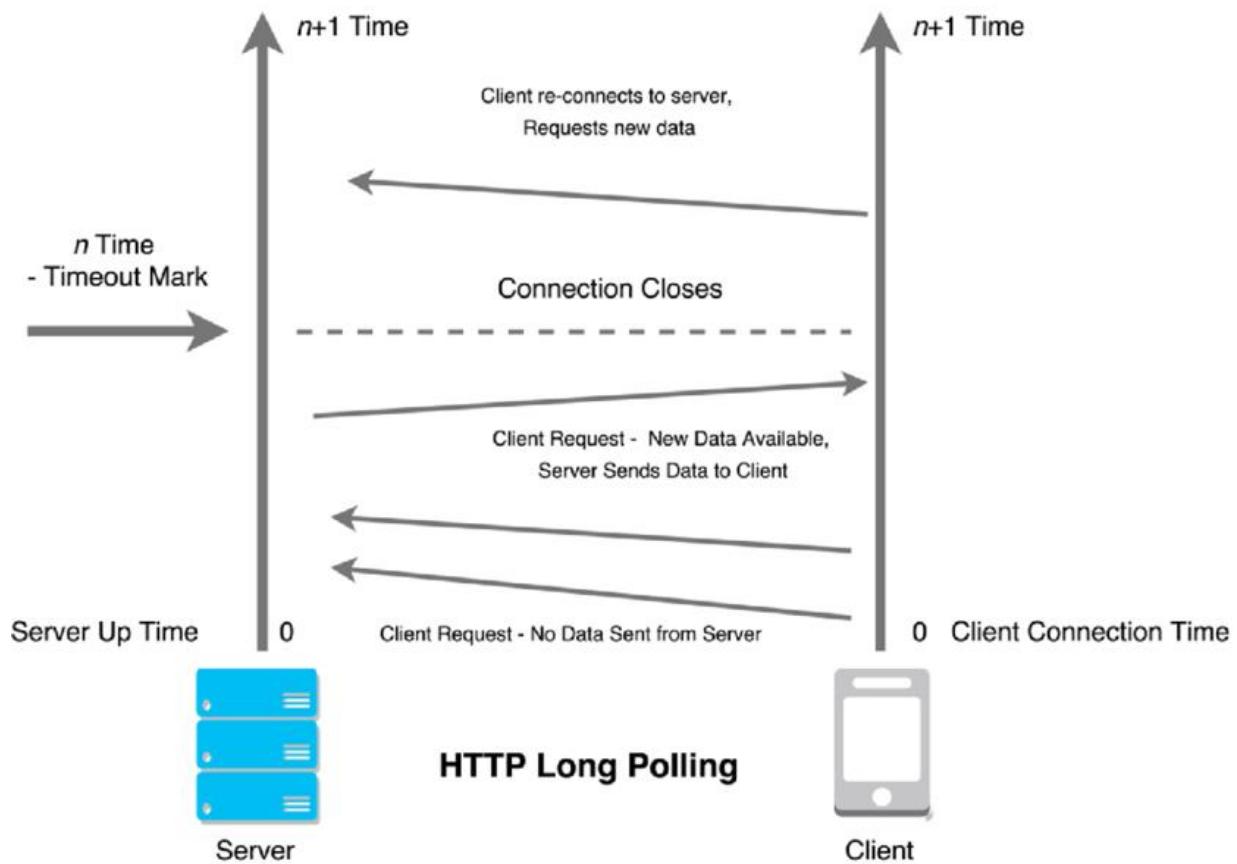


Figure 5-2. Long Polling

In conclusion, WebSocket provides the best solution to achieve real-time bidirectional communication efficiently. It didn't depend on certain browser or environment moreover it didn't waste bandwidth for necessary operations.

WebSocket and Socket

Before moving on we should devote a little time to clarifying a misconception that often arises when talking about WebSocket and its relation with Socket. In fact, they are totally different things, Socket is the low-level mechanism that internet use on the world wide web. Sockets work at the upper layers of the Internet protocol (IP) stack, known as the transport layer, where data is passed from an application down to the network via the operating system.

When an application on the computer wishes to send and receive data from a network connection, it asks the operating system to open an Internet socket. The

socket is set up consisting of the protocol information, such as user datagram protocol (UDP) or transmission control protocol (TCP), as well as the sending and receiving addresses of both computers and the IP port number for the connection. It is also possible for an Internet socket to be created that bypasses the operating system and sends the raw packets without first letting the computer's operating system deal with the additional socket information.

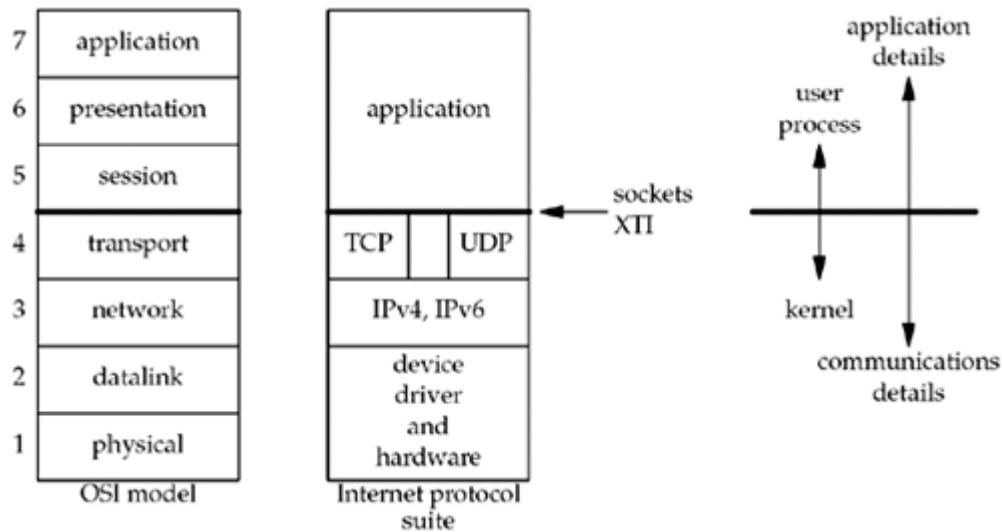


Figure 5-3. Network layers

On the other hand, WebSockets are more than just plain sockets. They use a framing protocol that requires a handshake and then exchanges messages masked by XORing them with a 32bit random number.

The reason for this additional encoding layer is that allowing a web browser to create arbitrary socket connections would open various security problems. You could, for example, make visitors to your website connect to arbitrary mail servers via SMTP and make them send spam without the user realizing it. That's why the protocol was designed in a way that any server-sided applications need to implement it intentionally before they can be used by web browsers.

We can summarize this section on the following image:

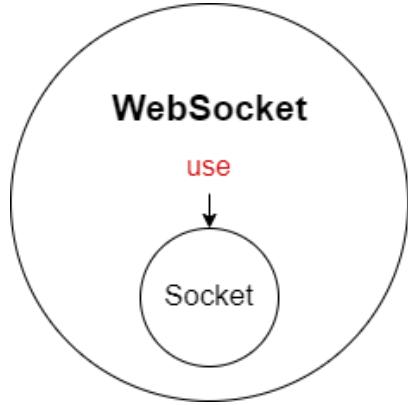


Figure 5-4. WebSoket and Socket

5.4 SocketIO

Socket.IO is JavaScript library built on top of WebSocket and other technologies. it uses WebSockets when available, but it's ready to fall back to other technologies such as Flash Socket, AJAX Long Polling, AJAX Multipart Stream, and many more; that allows Socket.IO to be used in contexts where WebSockets are not supported.

supported.



Figure 5-5.WebSocket

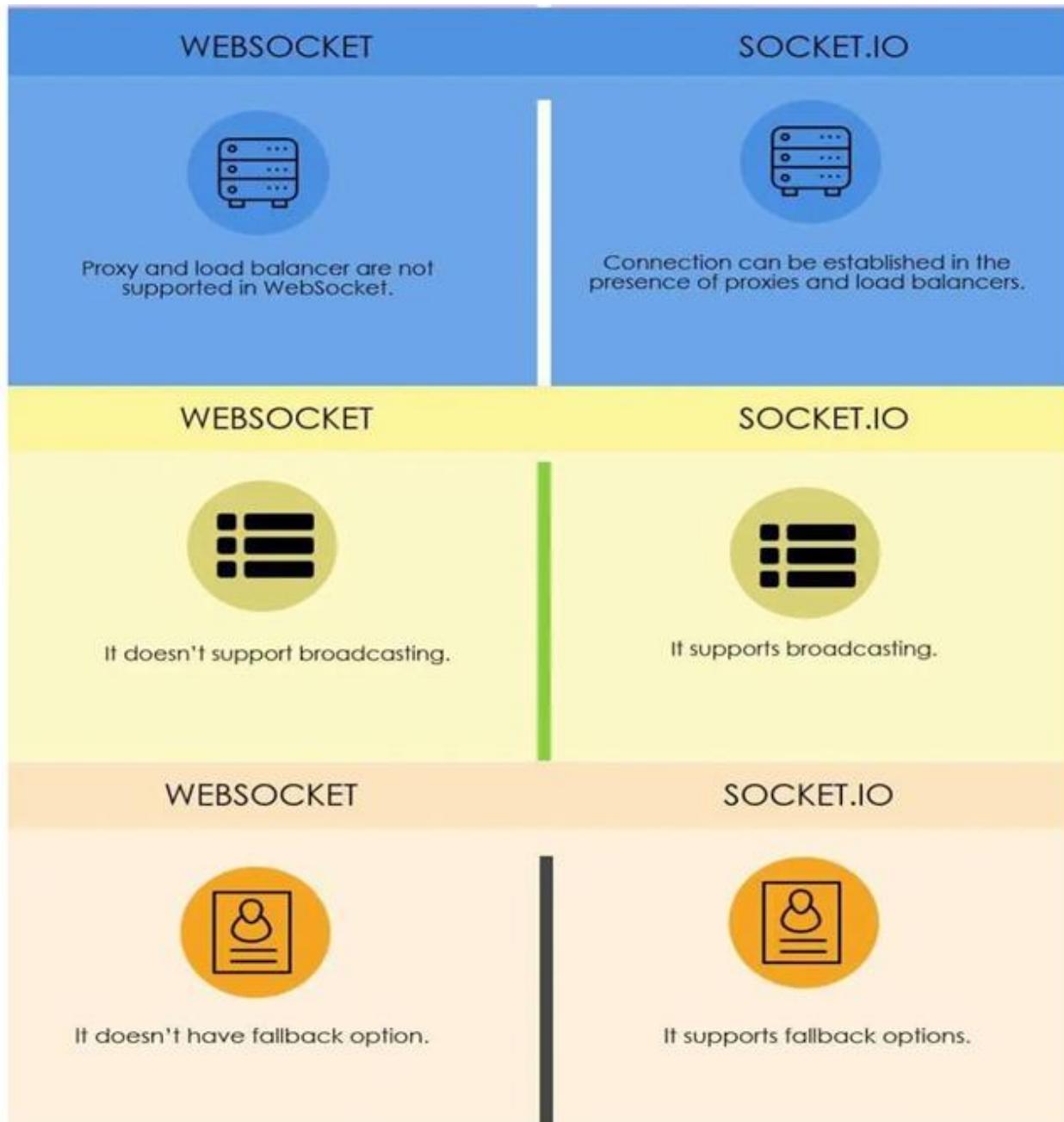


Figure 5-6. comparison between WebSocket and Socket.IO

So if Socket.IO is perfect why not use it instead of WebSocket?

As with all tech, choosing the right one means being clear on your ambitions for your product. Socket.IO does make many things easier in comparison to setting up

sockets yourself, but there are limitations and drawbacks in addition to the scaling issue mentioned above.

The first is that the initial connection is longer compared to WebSockets. This is due to it first establishing a connection using long polling and xhr-polling, and then upgrading to WebSockets if available.

If you don't need to support older browsers or GUI and aren't worried about client environments that don't support WebSockets you may not want the added overhead of Socket.IO. You can minimize this impact by specifying to only connect with WebSockets. This will change the initial connection to WebSocket but remove any fallback.

5.5 XMPP

After we have chosen web standard that we will use in our Smart Classroom we need to choose suitable protocol for communication between the server and the other nodes in the system. One of the best-known communication protocol is **XMPP** which is short for Extensible Messaging & Presence Protocol open standard that supports near-real-time chat and instant messaging by governing the exchange of XML data over a network.

XML, or Extensible Markup Language, provides a framework for storing and organizing plain text data within documents so that the data can be easily interpreted by a wide variety of network endpoints regardless of their hardware or software configuration. XMPP allows XML data, in the form of short snippets called stanzas, to be reliably sent from one endpoint to another using the internet's Transmission Control Protocol (TCP), passing through an intermediary server along the way. XMPP has many standards that organize their operator, The core one is rfc6120 which describes the main functionality of the protocol.

Functional summary of XMPP

The purpose of XMPP is to enable the exchange of relatively small pieces of structured data (called "XML stanzas") over a network between any two (or more) entities. XMPP is typically implemented using a distributed client-server architecture, wherein a client needs to connect to a server in order to gain access to the network and thus be allowed to exchange XML stanzas with other

entities (which can be associated with other servers). The process whereby a client connects to a server, exchanges XML stanzas, and ends the connection is:

Within XMPP, one server can optionally connect to another server to enable inter-domain or inter-server communication. For this to happen, the two servers need to negotiate a connection between themselves and then exchange XML stanzas; the process for doing so is:

- 1 - Determine the IP address and port at which to connect, typically based on the resolution of a fully qualified domain name
- 2 - Open a TCP connection
- 3 - Open an XML stream
- 4 - Preferably negotiate TLS for channel encryption
- 5 - Authenticate using a Simple Authentication and Security Layer [SASL] mechanism
- 6 - Exchange an unbounded number of XML stanzas both directly for the servers and indirectly on behalf of entities associated with each server, such as connected clients
- 7 - Close the XML stream
- 8 - Close the TCP connection

5.4.1 How Whatsapp use XMPP

One of the famous applications that use XMPP standard is Whatsapp but they spend great efforts at customizing and editing the standard to meet their requirements giving a good example that we will follow in our project.

FunXMPP

WhatsApp uses a protocol which is a slimmed-down version of XMPP but as we have said before it is a messaging protocol using XML as its syntax. A simple example of an XMPP message would be:

```
<message to="34123456789@s.whatsapp.net" type="text" id="message-1417651059-2" t="1417651059">
  <body>Test</body>
</message>
```

But apparently the creators of WhatsApp thought this was too bloated and found a way to express XMPP messages using only a few bytes, which they called FunXMPP. Since WhatsApp is intended for mobile devices which often lack a good internet connection, it is logical they wanted as few overhead as possible. Using FunXMPP they achieved that, while still using a standard internet protocol.

So how does FunXMPP accomplish this?

First of all, all keywords are assigned a byte. In the above example there are a lot of keywords that are common in xmpp (eg message, from, type, text).

If you can replace those with just one byte, it would reduce a lot of overhead. FunXMPP uses a HashTable for this, containing most (if not all) keywords.

Given the syntax \xnn for one byte with the hexadecimal value nn, the above example could be reduced to:

```
<\x59 \xa5="01234567890@\x91" \xa7="\xa2" \x44="message-1417651059-2"
\xa1="1417651059">
<\x12>Test</\x12>
</\x58>
```

Keeping in mind that \xnn stands for just one byte, this is already a significant reduce in size. Note that all remaining ascii values (eg 1417651059, Test, message-1417651059-2) cannot be replaced by anything because they are variable (ie can be set by the user).

XML is a human readable format employing tags that must be opened and closed. Is this really necessary for a computer to read an XML structure? The creators of FunXMPP must have thought the same thing because the other method of decreasing the size of messages is the encoding of the XML structure as a few bytes.

The only thing that remains now is the XML structure. In FunXMPP this structure is expressed as a set of lists. A list is designated by a \xf8 byte. After this \xf8 byte

comes a byte with the number of items the list contains. Things that count as one item here are: the tag name, keys, values and the body.

In general: a list followed directly by a list means there are several nodes at the same level and the first list is not a tag or anything visible in the XML.

Token MapLookup

```
0x03 => 'account',
0x04 => 'ack',
0x05 => 'action',
0x06 => 'active',
0x07 => 'add',
0x08 => 'after',
0x09 => 'all',
0x0a => 'allow',
0x0b => 'apple',
0x0c => 'auth',
0x0d => 'author',
0x0e => 'available',
0x0f => 'bad-protocol',
0x10 => 'bad-request',
0x11 => 'before',
0x12 => 'body',
0x13 => 'broadcast',
0x14 => 'cancel',
0x15 => 'category',
0x16 => 'challenge',
0x17 => 'chat',
0x18 => 'clean',
```

Example of WhatsApp Registration Flow

WhatsApp registration process is very confusing or no official document is available for guidance. Request tracking using traffic monitoring tools has revealed following truth about registration process. Below is the simplified version of process flow.

Understanding the WhatsApp Registration Flow

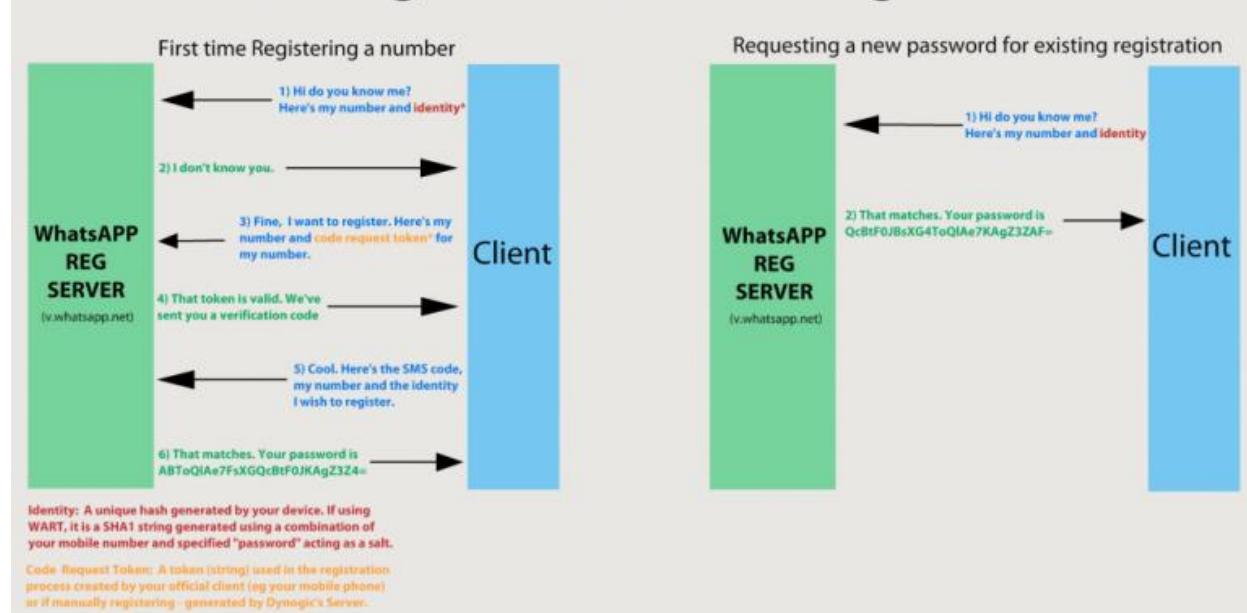


Figure 5-7. WhatsApp Registration Flow.

XMPP in Smart Classroom

The first customization we made to XMPP protocol is that we use JSON file base instead of XML. JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate.

It is based on a subset of the JavaScript Programming Language Standard ECMA-262 3rd Edition - December 1999. JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language.

XML

VS.

JSON

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <endereco>
3   <cep>31270901</cep>
4   <city>Belo Horizonte</city>
5   <neighborhood>Pampulha</neighborhood>
6   <service>correios</service>
7   <state>MG</state>
8   <street>Av. Presidente Antônio Carlos, 6627</street>
9 </endereco>
```

```

1- {
2-   "endereco": {
3-     "cep": "31270901",
4-     "city": "Belo Horizonte",
5-     "neighborhood": "Pampulha",
6-     "service": "correios",
7-     "state": "MG",
8-     "street": "Av. Presidente Antônio Carlos, 6627"
9-   }
10 }
```

The biggest advantage of JSON over XML is that XML has to be parsed with an XML parser. JSON can be parsed by natively by any programming language and you don't have to write custom logic to parse it.

Another advantage is the size, XML has a much larger size due to repeated and unnecessary tags that do not exist in JSON that results in huge boast in performance and network bandwidth. Next, we will discuss the customized connection sequence that we applied.

Main connection sequence

The client has to perform the following steps to utilize the server operations:

- 1- Use IP address and port to connect the socket server.
- 2- perform a two-way handshake using json encoded instructions.
- 3- wait for commands from the server in a JSON sanitized format.
- a - when a command came, perform the operations requested from the server.
- b - send message to the server containing the results of the operation over JSON sanitized format.
- 4- send Commands to the server in a JSON sanitized format.
- 6- close the connection.

In the next few lines we will discuss each of these operations in a more details

Operation 1

The client needs to connect to the server using IP address and Port these may be fixed system or dynamically changed. In the latter case, the client need to consult a DNS server to get the needed credentials.

Operation 2

The client needs to advertise itself to the server to inform it about its physical properties like device id and the client role. The server then check if the client had previously advertised itself if so return an error message, else identify the client then send it confirmation message.

Operation 3 && Operation 4

The two operation is main building block of the system. The server and the client can communicate to each other in real-time bidirectional tunnel. The clients can also use the server as a gateway to communicate with other clients either they have the same role or not.

5.5 WebSocket Implementation

So far we have discussed the theoretical aspect of WebSocket in our Smart Classroom project. Next we will take about how we implemented the WebSocket server using Ratchet PHP library. Ratchet is a loosely coupled PHP library providing developers with tools to create real time, bi-directional applications between clients and servers over WebSockets.



Ratchet main features

- 1- Open-source:
Ratchet is community driven not proprietary of a certain company or enterprise. It is also free compared to other paid solutions.
- 2- Event-driven:
After understanding "the new flow" - event driven programming, compared to traditional HTTP request/response - writing any application on top of Ratchet becomes fast and easy.
- 3- Structured Component: The core of Ratchet is made up of Components. Each component implements a version of the Component Interface. If you follow that link you can see each of the classes that implement ComponentInterface.

Each class is instantiated when the script is launched, then enters an event loop, where I/O listens and calls the class on top of it. (it does not trigger a global event, it passes the event on to one class attached to [below] it).

An event is triggered at the top of the table (seen below) from a client on the other side of the socket. The client connection associated with the event then propagates up the structure along with any information sent.

Each class defines which interface it accepts, then propagating its own events. This structure allows developers to add or subtract class components to create different functionality. For example, one may want to add a logging component between WebSocket and WAMP to log raw JSON messages received by the client.

Below is an example of an Application put together using various components. You can see how by adding more layers, Components are able to extend and further define raw data into more specific events. As seen below WAMP accepts a data event, it then parses that data (JSON) and propagates its own events based on the data received.

Component Class	Event triggered by Client (JavaScript)								
I/O (socket transport)	open	close	data						error
HTTP Protocol Handler	open	close	data						error
WebSocket Protocol Handler	open	close	data						error
Session Provider	open	close	data						error
WAMP Protocol Handler	open	close	publish	subscribe	unsubscribe	call	prefix		error
(your) Application	open	close	publish	subscribe	unsubscribe	call	prefix		error

Message Format

The standard JSON message format is as follow:

```
{  
  "action" : "xxxxx",  
  "to" : "xxxxx",  
  "from" : "xxxxx",  
  "device_id": "xxxxx",  
  "execute" : {  
  }  
}
```

If we compare it to the standard XMPP XML such as

```
<message to="34123456789@s.whatsapp.net" type="text" id="message-1417651059-2" from="1417651059">  
  <body>Test</body>  
</message>
```

We will find out that the fields form, to, device_id correspond to from, to, id respectively, whereas action act as type and execute as the body.

The actual value of these fields is related directly to the application but first, let us devote more time at the details of system design.

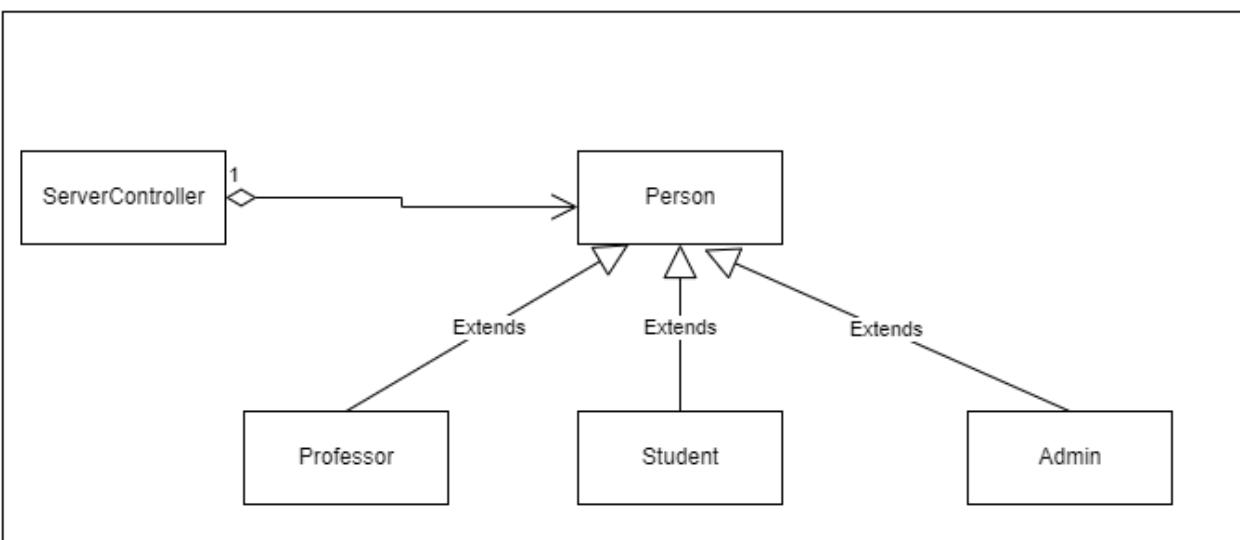
The users of the application fall into three categories students, professors, admins. Admin should use Websocket server to add new students into the system. Professor should use the server to verify the students record attendance, make tests and manipulate other information in the session. Students are the main user of the system, Our Smart Classroom target them mainly to enhance their education experience.

Software design and object-oriented

As we mentioned earlier Ratchet is event-based system so when message hit the system we redirect it handle_message method of the ServerController object.

```
public function onMessage(ConnectionInterface $from, $msg) {  
    $this->controller->handleMsg( $from, $msg);  
}
```

After that we validate the message if it's correct we proceed into inspecting the message action type and perform the necessary operations. Here is the object UML diagram:



As you can see the Person provide the common properties and methods that are inherited by Student, Professor, Admin. SeverController contain lists of these objects to validate them and handle their operation.

The following section contains object design artifacts

System use case diagrams

In the Unified Modeling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system.

the use case usually contains:

- Scenarios in which the system or application interacts with people, organizations, or external systems
- Goals that your system or application helps those entities (known as actors) achieveThe scope of your system

Professor USE CASE

1. Professor Login to the system
2. if login failed go to step 1
3. Professor connect to the websocket server and provide his login token
4. If The professor try to connect again, he get an error.
5. Professor ask for the list of students in the class.
6. Professor ask to verify these students and get a list of verified student.
7. Professor can verify single student.
8. professor can send files and attachment.
9. Professor can send exams/quiz to the student.
10. Professor submit the session result to the database.

Admin USE CASE

1. Admin Login to the system
2. if login failed go to step 1
3. Admin connect to the websocket server and provide his login token
4. If The admin try to connect again, he gets an error.
5. Admin ask for the list of students in the class that need registration.
6. Admin ask to begin the recognition phase.
7. Admin submit the session result to the database.

Student USE CASE

1. Student access his/her device
2. Student wait for
3. Admin connects to the websocket server and provide his login token
4. If The admin tries to connect again, he gets an error.
5. Admin asks for the list of students in the class that need registration.
6. Admin asks to begin the recognition phase.
7. Admin submits the session result to the database.

5.6 Conclusion and future considerations

In these pages we had discussed our core features of Our Smart Classroom server and the internal implementation of its different components and the multiple tech stack and protocol that power them. However we are finished yet there are missing features that will push our project forward.

One of the important features is Data-encryption helps protect private information, sensitive data, and can enhance the security of communication between client apps and servers.

Unfortunately, PHP doesn't provide built-in easy solutions to encrypt data like other languages like Java, so we have to use a community-made library. A good candidate is libsodium.

Sodium is a new, easy-to-use software library for encryption, decryption, signatures, password hashing, and more.

It is a portable, cross-compilable, installable, packageable fork of NaCl, with a compatible API, and an extended API to improve usability even further.

Its goal is to provide all of the core operations needed to build higher-level cryptographic tools.

6. Chapter six: Backend application

6.1 Introduction:

This term is used widely when it comes to discussions about data. In databases, information is stored in tables, columns and rows for easy processing. That storage is managed by the DBMS – database management system. There are relational (SQL) and non-relational (NoSQL) databases. A relational database is generally said to be the most common kind. This is a type of database which enables users to access data presented in relation to other pieces of data in the same storage.

A different type of database is the operational data store (ODS) – it provides a snapshot of the latest data for the user from many transactional systems. This can be then used for preparing operational reports. The ODS will allow your company to gather up-to-date data from many sources in its original format for reporting purposes.

Is this the best way for your business to store data? You could invest in databases instead of other solutions if quick and frequent access to data is what you value most. If you work in an industry which relies on data and requires its fast processing (e.g. transport, like airlines or train networks, which have to process tickets online), a database is a good choice.

A Data Warehouse is a multi-purpose system – it can be used by various companies in any industry. What is special about it, is that the data warehouse can retrieve information from various sources in your organization. This type of software is capable of collecting data from numerous applications for later storage and management. As the data is copied from the source into the data warehouse it can be immediately analyzed.

Data warehouses usually use SQL in order to query the data, exactly like databases. Data is stored in tables and organized by types with keys, indexes etc. and can be used for creating reports or for analytical purposes. Data in warehouses is updated and users can check how data is changed.

Data mart is a repository for summarized, subject-oriented data from the data warehouse. This solution may suit specific users and teams within your

organization, although it is hardly a type of tech that could serve all of your employees. A data mart allows some employees to access specific types of data stored in the data warehouse, among other kinds – for example, accountants can use it to get financial data and marketing specialists can gain data needed to create a new strategy. In short, it is designed and developed for particular business tasks.

There are 3 types of Data Marts:

- Independent Data Mart – which doesn't rely on an existing data warehouse. Data is taken from original sources when analysis is needed.
- Dependent Data Mart – is connected with an existing data warehouse and when users want to run analysis, they can access only specific pieces of data from the data warehouse.
- Hybrid Data Mart – gets the data from external operational sources and data warehouses.

All kinds of raw data generated in your organizations can be stored in a data lake, so any user that has been granted access can retrieve relevant, needed data. This storage is done without any processing though – each piece of information is like a small drop in the lake. Everything is in there – structured data, chat logs, e-mails, images, documents, videos. So what is the difference between data warehouses and data lakes as they can both store large amounts of data? A warehouse stores processed data, while a data lake stores raw data as well.

What is the problem with data lakes? As they grant more flexibility to a user, they're also less secure. A company may need a data scientist to use data lakes. They may be useful though for big organizations that have to analyze a great deal of data to improve their business.

There are different ways to store and process data. Everything depends on what you need. Some data storing solutions can be better for you and others may cost you a lot of time and money.

6.2 Access and modify this data

Database

A database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a database management system (DBMS). Together, the data and the DBMS, along with the applications that are associated with them, are referred to as a database system, often shortened to just database.

A database management system is a software package for creating and managing databases. Many different types of database systems exist based on how they manage the database structure.

Two types of database structure

Databases typically have one of two basic forms:

- single-file or flat file database
- multi-file relational or structured database

· A flat file database stores data in a plain text file, with each line of text typically holding one record. Delimiters such as commas or tabs separate fields. A flat file database uses a simple structure and, unlike a relational database, cannot contain multiple tables and relations.

· A relational database contains multiple tables of data with rows and columns that relate to each other through special key fields. These databases are more flexible than flat file structures, and provide functionality for reading, creating, updating, and deleting data. Relational databases use Structured Query Language (SQL) - a standard user application that provides an easy programming interface for database interaction.

6.3 Relational database

A relational database (RDB) is a collective set of multiple data sets organized by tables, records and columns. RDBs establish a well-defined relationship between database tables. Tables communicate and share information, which facilitates data searchability, organization and reporting.

RDBs use Structured Query Language (SQL), which is a standard user application that provides an easy programming interface for database interaction.

RDB is derived from the mathematical function concept of mapping data sets and was developed by Edgar F. Codd.

Techopedia Explains Relational Database (RDB)

RDBs organize data in different ways. Each table is known as a relation, which contains one or more data category columns. Each table record (or row) contains a unique data instance defined for a corresponding column category. One or more data or record characteristics relate to one or many records to form functional dependencies. These are classified as follows:

- One to One: One table record relates to another record in another table.
- One to Many: One table record relates to many records in another table.
- Many to One: More than one table record relates to another table record.
- Many to Many: More than one table record relates to more than one record in another table.

RDB performs "select", "project" and "join" database operations, where select is used for data retrieval, project identifies data attributes, and join combines relations.

RDBs have many other advantages, including:

- Easy extendability, as new data may be added without modifying existing records. This is also known as scalability.
- New technology performance, power and flexibility with multiple data requirement capabilities.
- Data security, which is critical when data sharing is based on privacy. For example, management may share certain data privileges and access and block employees from other data, such as confidential salary or benefit information

Four types of database management systems

A relational database management system is one of four common types of systems you can use to manage your business data. The other three include:

1. hierarchical database systems
2. network database systems
3. object-oriented database systems
4. Hierarchical database model resembles a tree structure, similar to a folder architecture in your computer system. The relationships between records are pre-defined in a one to one manner, between 'parent and child' nodes. They require the user to pass a hierarchy in order to access needed data. Due to limitations, such databases may be confined to specific uses.

Network database models also have a hierarchical structure. However, instead of using a single-parent tree hierarchy, this model supports many to many relationships, as child tables can have more than one parent.

Finally, in object-oriented databases, the information is represented as objects, with different types of relationships possible between two or more objects. Such databases use an object-oriented programming language for development.

6.4 Non-relational database

A popular alternative to relational databases, NoSQL databases take a variety of forms and allow you to store and manipulate large amounts of unstructured and semi-structured data. Examples include key-value stores, document stores and graph databases.

A non-relational database is a database that does not incorporate the table/key model that relational database management systems (RDBMS) promote. These kinds of databases require data manipulation techniques and processes designed to provide solutions to big data problems that big companies face. The most popular emerging non-relational database is called NoSQL (Not Only SQL).

Most non-relational databases are incorporated into websites such as Google, Yahoo!, Amazon and Facebook. These websites introduce a slew of new applications every single day with millions and millions of users, so they would not be able to handle large traffic spikes with existing RDBMS solutions. Since RDBMS cannot handle the problem, they've switched to a new kind of DBMS that

is capable of handling Web scale data in a non-relational way. An interesting aspect of a non-relational database such as NoSQL is scalability. NoSQL uses the BASE system (basically available, soft-state, eventually consistent). Non-relational databases forgo the table form of rows and columns relational databases use in favor of specialized frameworks to store data, which can be accessed by special query APIs. Persistence is an important element in these databases. To enable fast throughput of vast amounts of data the best option for performance is "in memory," rather than reading and writing from disks. Relational databases use the ACID system, which ensures consistency of data in all situations of data management but obviously takes longer to process because of all those relations and its branching nature. However, the BASE system loosened up the requirements on consistency to achieve better availability and partitioning for better scalability.

When constructing the backend of an application, you need to take into account how the frontend will talk to the backend. More important, however, is the construction and design of your database. The relationships your data forms will lead to the construction of your database schema.

6.5 Database schema is an abstract design that represents the storage of your data in a database. It describes both the organization of data and the relationships between tables in a given database. Developers plan a database schema in advance so they know what components are necessary and how they will connect to each other.

In this guide, we will learn what a database schema is and why they are used. We will go through a few common examples so you can learn how to configure a database schema on your own.

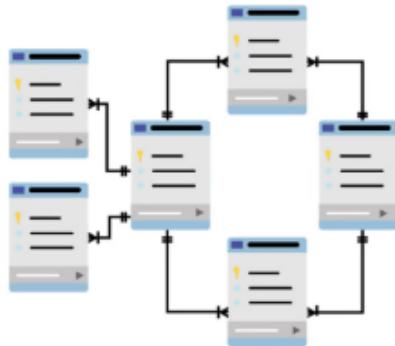


Figure 6-1. Database schema

When it comes to choosing your database, one of the things you have to think about is the shape of your data, what model it will follow, and how the relationships formed will help us as we develop a schema.

A database schema is a **blueprint** or **architecture** of how our data will look. It doesn't hold data itself, but instead describes the shape of the data and how it might relate to other tables or models. An entry in our database will be an instance of the database schema. It will contain all of the properties described in the schema.

Think of a database schema as a type of data structure. It represents the framework and arrangement of the contents of an organization's data.

A database schema will include:

- All important or relevant data
- Consistent formatting for all data entries
- Unique keys for all entries and database objects
- Each column in a table has a name and data type

The size and complexity of your database schema depends on the size of your project. The visual style of a database schema allows programmers to structure

the database and its relationships properly before jumping into the code. The process of planning a database design is called data modeling.

Schemas are important for designing database management systems (DBMS) or relational database management systems (RDBMS). A DBMS is a software that stores and retrieves user data in a secure way that follows the ACID concept.

In many companies, database design and DBMS responsibilities usually fall to the role of the **Database Administrator (DBA)**. DBAs are responsible for ensuring that data analysts and database users can easily access information. They work alongside management teams to plan and securely manage an organization's database.

Note: Some popular DBMS systems are MySQL, Oracle, PostgreSQL, Microsoft Access, MariaBB, and dBASE, amongst others.

Database schema types

There are two main database schema types that define different parts of the schema: **logical** and **physical**.

Logical

A logical database schema represents **how the data is organized** in terms of tables. It also explains how attributes from tables are linked together. Different schemas use a different syntax to define the logical architecture and constraints.

Note: Integrity constraints are a set of rules for a DBMS that maintain quality for data insertion and updates.

To create a logical database schema, we use tools to illustrate relationships between components of your data. This is called entity-relationship modeling (ER Modeling). It specifies what the relationships between entity types are.

The schema diagram below is a very simple ER Model that shows the logical flow in a basic commerce application. It explains a product to a customer who buys a product.

Database management architecture (DBMS):

- DBMS Architecture (Three Schema Architecture)

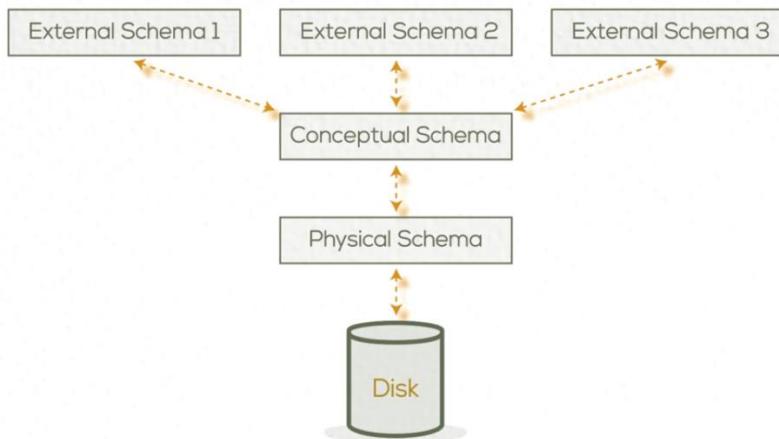


Figure 6-2: DBMS Architecture

1. External Schema: They are concerned with what data user will see and how the data will be presented to the user.
 2. Conceptual Schema or The logical model: They are concerned with what is represented. (define database structures such as tables and constraints)
 3. Physical Schema or The physical model: How the data are represented in the database? And How the data structures are implemented?
- Mapping : it is the processes of transforming requests and results between levels of DBMS Architecture.
 - Data model

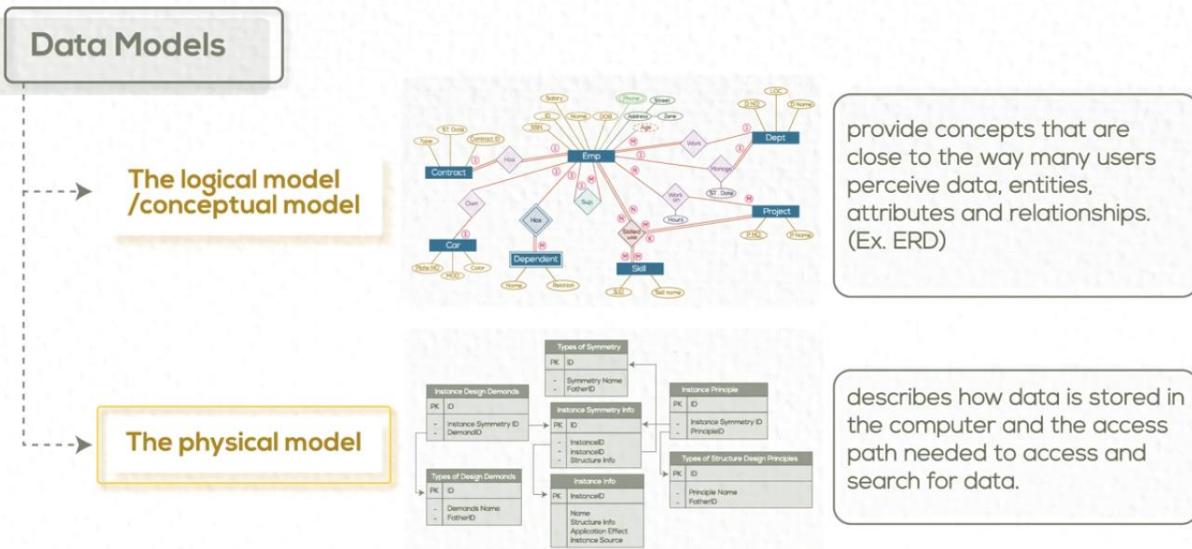


Figure 6-3: Data model

- **DBMS functions:** types that DBMS can support (store and retrieve)
 1. **Text/Number/Image/Audio/Video (Multimedia Function).**
 2. **Spatial Data.**

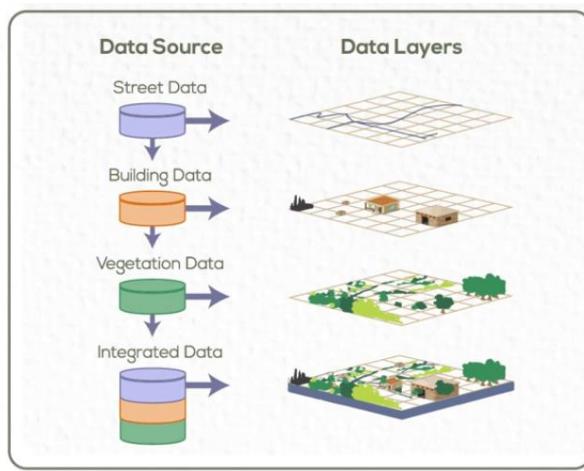


Figure 6-4. Data

3. **Time Series (Data versus Time).**

4. Data Mining: can make analysis for the data by applying Clustering, Classification and Association Rules.

- **Database environment.**

➤ **Centralized Database Environment:**

1. Mainframe environment: Database server and application server are in the same machine.

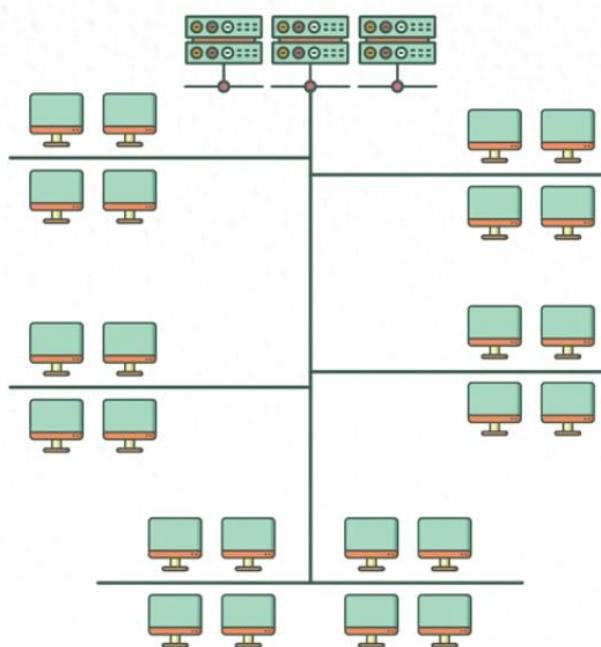


Figure 6-5.Mainframe environment

Problem with this environment

- The processing depends on one server.
- The performance is very slow.
- Database and application layer has Single Point of failure.

2. Client/Server environment: two environment Database server and application sever has been installed and setup at the client in this case we call the client by thick client because Application locally installed.



Figure 6-6. Client/Server environment

Problem with this environment

- Database is a single point of failure.
- High cost for support.

Advantages

- Application layer is not a single point of failure.

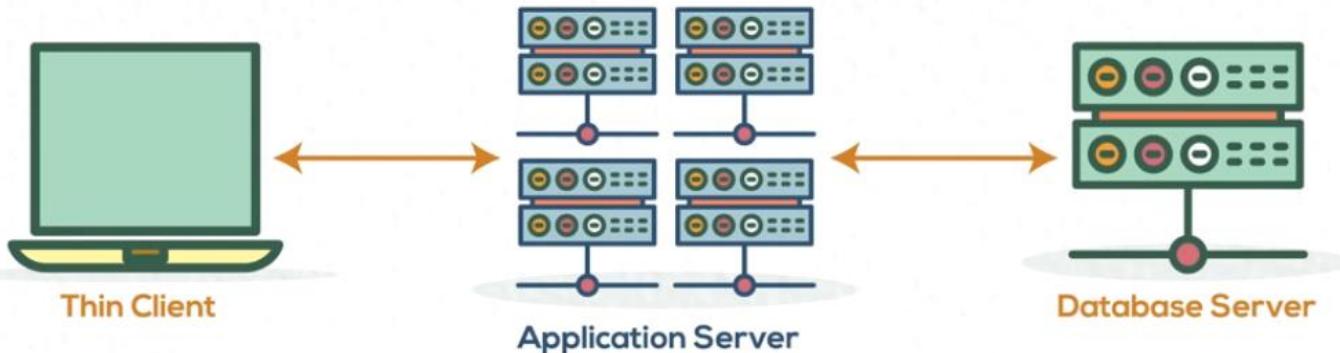


Figure 6-7. Client/Server environment

3. Internet computing environment (three-tier environment): in this case we can call client by thin client because it can access Application by using browser to access application server.

Problem with this environment

- Application server is a single point of failure.
- Database is a single point of failure.

Advantages

- Lower cost for support and maintenance.

Now let's get into our project database implementation part:

In our project database model we have a lot of entities like **Students, Professors, Lectures** and others

These entities should have **attributes** to describe these entities like **Student ID, Professor ID, Lecture ID** and others

The physical model for our project data base can be described as following model.

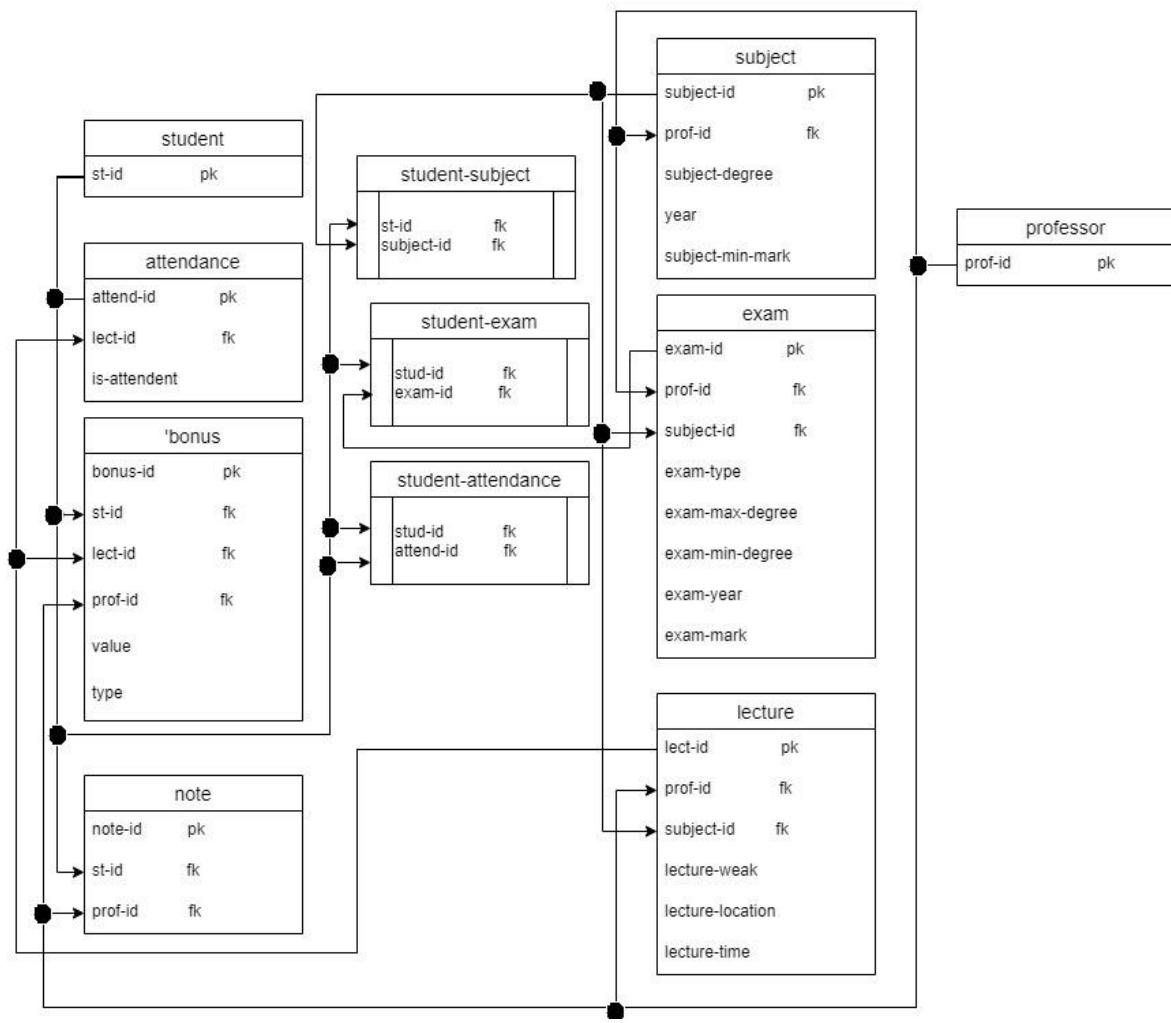


Figure 6-8. physical model for our project

So as we see our project database model is **RELATIONAL DATABASE**
So we can deal with Structured Query language (SQL) pronounced as “S-Q-L”

Or sometimes as “see-Quel” is actually the standard language for dealing with Relational Databases.

6.6 DBMS:

Is a software that stores and retrieves user data in a secure way that follows the ACID concept.

SQL CAN BE EFFECTIVELY USED TO ...

INSERT, SEARCH, UPDATE, DELETE, DATABASE RECORDS.

BUT that doesn't mean SQL cannot do things beyond that. In fact it help in optimizing and maintenance of databases and much more.

Relational databases like MYSQL, ORACLE, MS SQL SERVER, SYBASE, etc uses SQL. SQL SYNTAXES used in these databases are almost similar, except the fact that some databases use different syntaxes and even proprietary SQL syntaxes.

EXAMPLE OF SQL STATEMENT

Select * from members where age >30

- DBMS stands for database management system.
- We have four major types of DBMSs namely hierarchical, network, relational, object oriented.
- The most widely used DBMS is the relational model that saves data in table formats.
- It uses SQL ass the standard query language used to query a database.
- The database approach has many advantages when it comes to storing data compared to the traditional flat file based systems.

applications enabled by the explosion of digitized information over the last five years will require the solution to significant new research problems. These problems are grouped in this report into the following broad areas: support for multimedia objects, distribution of information, new database applications, workflow and transaction management, and ease of database management and use. A new research mandate for the database community is provided by the technology developments of the recent past - the explosions in hardware capability, hardware capacity, and communication (including the internet or "web" and mobile communication).

Prerequisites – SQL, NoSQL

When it comes to choosing a database the biggest decisions is picking a relational (SQL) or non-relational (NoSQL) data structure. While both the databases are viable

options still there are certain key differences between the two that users must keep in mind when making a decision.

The Main Differences:

1. **Type – SQL** databases are primarily called as Relational Databases (RDBMS); whereas NoSQL database are primarily called as non-relational or distributed database.

2. **Language – SQL** databases defines and manipulates data based structured query language (SQL). Seeing from a side this language is extremely powerful. SQL is one of the most versatile and widely-used options available which makes it a safe choice especially for great complex queries. But from other side it can be restrictive. SQL requires you to use predefined schemas to determine the structure of your data before you work with it. Also all of your data must follow the same structure. This can require significant up-front preparation which means that a change in the structure would be both difficult and disruptive to your whole system.

A NoSQL database has dynamic schema for unstructured data. Data is stored in many ways which means it can be document-oriented, column-oriented, graph-based or organized as a KeyValue store. This flexibility means that documents can be created without having defined structure first. Also each document can have its own unique structure. The syntax varies from database to database, and you can add fields as you go.

3. **Scalability** – In almost all situations SQL databases are vertically scalable. This means that you can increase the load on a single server by increasing things like RAM, CPU or SSD. But on the other hand NoSQL databases are horizontally scalable. This means that you handle more traffic by sharding, or adding more servers in your NoSQL database. It is similar to adding more floors to the same building versus adding more buildings to the neighborhood. Thus NoSQL can ultimately become larger and more powerful, making these databases the preferred choice for large or ever-changing data sets.

4. **Structure** – SQL databases are table-based on the other hand NoSQL databases are either key-value pairs, document-based, graph databases or wide-column stores. This makes relational SQL databases a better option for applications that require multi-row transactions such as an accounting system or for legacy systems that were built for a relational structure.

5. **Property followed** – SQL databases follow ACID properties (Atomicity, Consistency, Isolation and Durability) whereas the NoSQL database follows the Brewers CAP theorem (Consistency, Availability and Partition tolerance).

6. **Support** – Great support is available for all SQL database from their vendors. Also a lot of independent consultations are there who can help you with SQL database for a very large scale deployments but for some NoSQL database you still have to rely on community support and only limited outside experts are available for setting up and deploying your large scale NoSQL deployments.

6.7 Back-End Web Architecture

The front-end is the code that is executed on the client side. This code (typically HTML, CSS, and JavaScript) runs in the user's browser and creates the user interface.

The back-end is the code that runs on the server, that receives requests from the clients, and contains the logic to send the appropriate data back to the client. The back-end also includes the database, which will persistently store all of the data for the application. This article focuses on the hardware and software on the server-side that make this possible.

Review [HTTP](#) and [REST](#) if you want to refresh your memory on these topics. These are the main conventions that provide structure to the request-response cycle between clients and servers.

What are the clients?

The clients are anything that send requests to the back-end. They are often browsers that make requests for the HTML and JavaScript code that they will execute to display websites to the end user. However, there are many different kinds of clients: they might be a mobile application, an application running on another server, or even a web enabled smart appliance.

What is a back-end?

The back-end is all of the technology required to process the incoming request and generate and send the response to the client. This typically includes three major parts:

The server. This is the computer that receives requests.

The app. This is the application running on the server that listens for requests, retrieves information from the database, and sends a response.

The database. Databases are used to organize and persist data.

What is a server?

A server is simply a computer that listens for incoming requests. Though there are machines made and optimized for this particular purpose, any computer that is connected to a network can act as a server. In fact, you will often use your very own computer as server when developing apps.

What are the core functions of the app?

The server runs an app that contains logic about how to respond to various requests based on the [HTTP verb](#) and the [Uniform Resource Identifier \(URI\)](#). The pair of an HTTP verb and a URI is called a route and matching them based on a request is called routing.

Some of these handler functions will be middleware. In this context, middleware is any code that executes between the server receiving a request and sending a response. These middleware functions might modify the request object, query the database, or otherwise process the incoming request. Middleware functions typically end by passing control to the next middleware function, rather than by sending a response.

Eventually, a middleware function will be called that ends the request-response cycle by sending an HTTP response back to the client.

Often, programmers will use a framework like Express or Ruby on Rails to simplify the logic of routing. For now, just think that each route can have one or many handler functions that are executed whenever a request to that route (HTTP verb and URI) is matched.

What kinds of responses can a server send?

The data that the server sends back can come in different forms. For example, a server might serve up an HTML file, send data as JSON, or it might send back only an [HTTP status code](#). You've probably seen the status code "404 - Not Found" whenever you've tried navigating to a URI that doesn't exist, but there are

many more status codes that indicate what happened when the server received the request.

What is a database, and why do we need to use them?

Databases are commonly used on the back-end of web applications. These databases provide an interface to save data in a persistent way to memory. Storing the data in a database both reduces the load on the main memory of the server CPU and allows the data to be retrieved if the server crashes or loses power.

Many requests sent to the server might require a database query. A client might request information that is stored in the database, or a client might submit data with their request to be added to the database.

What is a Web API, really?

An API is a collection of clearly defined methods of communication between different software components.

More specifically, a Web API is the interface created by the back-end: the collection of endpoints and the resources these endpoints expose.

A Web API is defined by the types of requests that it can handle, which is determined by the routes that it defines, and the types of responses that the clients can expect to receive after hitting those routes.

One Web API can be used to provide data for different front-ends. Since a Web API can provide data without really specifying how the data is viewed, multiple different HTML pages or mobile applications can be created to view the data from the Web API.

Other principles of the request-response cycle:

The server typically cannot initiate responses without requests!

Every request needs a response, even if it's just a 404 status code indicating that the content was not found. Otherwise your client will be left hanging (indefinitely waiting).

The server should not send more than one response per request. This will throw errors in your code.

Mapping out a request

by following an example of the main steps that happen when a client makes a request to the server.

1. Alice is shopping on SuperCoolShop.com. She clicks on a picture of a cover for her smartphone, and that click event makes a GET request to

<http://www.SuperCoolShop.com/products/66432>.

Remember, GET describes the kind of request (the client is just asking for data, not changing anything). The URI (uniform resource identifier) /products/66432 specifies that the client is looking for more information about a product, and that product, has an id of 66432.

SuperCoolShop has an huge number of products, and many different categories for filtering through them, so the actual URI would be more complicated than this. But this is the general principle for how requests and resource identifiers work.

2. Alice's request travels across the internet to one of SuperCoolShop's servers. This is one of the slower steps in the process, because the request cannot go faster than the speed of light, and it might have a long distance to travel. For this reason, major websites with users all over the world will have many different servers, and they will direct users to the server that is closest to them!

3. The server, which is actively listening for requests from all users, receives Alice's request!

4. Event listeners that match this request (the HTTP verb: GET, and the URI: /products/66432) are triggered. The code that runs on the server between the request and the response is called middleware.

5. In processing the request, the server code makes a database query to get more information about this smartphone case. The database contains all of the other information that Alice wants to know about this smartphone case: the name of the product, the price of the product, a few product reviews, and a string that will provide a path to the image of the product.

6. The database query is executed, and the database sends the requested data back to the server. It's worth noting that database queries are one of the slower steps in this process. Reading and writing from static memory is fairly slow, and the database might be on a different machine than the original server. This query itself might have to go across the internet!

7. The server receives the data that it needs from the database, and it is now ready to construct and send its response back to the client. This response body has all of the information needed by the browser to show Alice more details (price, reviews, size, etc) about the phone case she's interested in. The response header will contain an HTTP status code 200 to indicate that the request has succeeded.

8. The response travels across the internet, back to Alice's computer.

9. Alice's browser receives the response and uses that information to create and render the view that Alice ultimately sees!

serverless google cloud- microsoft azure - amazon cloud pros – cons server pro –cons why we choose server

6.8 Server vs. Serverless: Which Is the Best Option?

Even though serverless computing does still involve servers, the user is free of managing and monitoring them. You can delegate this task to a cloud provider and focus on other important tasks. This is one of the key reasons for the rapid popularity growth of serverless computing.

According to [2020 research by O'Reilly](#), 40% of respondents claim that their company has fully adopted serverless architecture. Half of these respondents say that the implementation was extremely successful.

So, what's so great about serverless computing, and why are some businesses still hesitant to make the change? In this article, we'll compare server vs. serverless, discuss the pros and cons of serverless architecture and server computing, and determine when it's most suitable to use one or the other.

What is a Server?

Server computing continues to remain the most reliable option in the minds of business owners. Many people prefer the traditional method, especially regarding security and data protection, as it's proven to be bulletproof. However, the increasing adoption rates of serverless computing indicate that server computing has its downsides. Let's explore the benefits and drawbacks of server computing and identify when the traditional option would be favorable.

Benefits and Disadvantages of Server Computing

Benefits of Server Computing

Full Access. Managed and maintained within the company by the in-house IT team, on-premise servers allow unlimited access. In other words, your IT department can install or alter any software, quickly fix issues, and have absolute control over backups.

Security. As all of your critical data is safely stored on the premises and no third-party service has access to it, the server computing model is known for high security. When debating server versus serverless, many people bring up this issue, and for a good reason. Data protection is an important part of IT activities, as breaches and hacker attacks are becoming more frequent and severe.

No dependency on internet connection. On-premise servers allow IT staff to access valuable data at any time, even when the internet connection is poor or unstable. Even in 2021, there are a lot of regions with disruptive internet service where people cannot safely rely on a stable connection at all times.

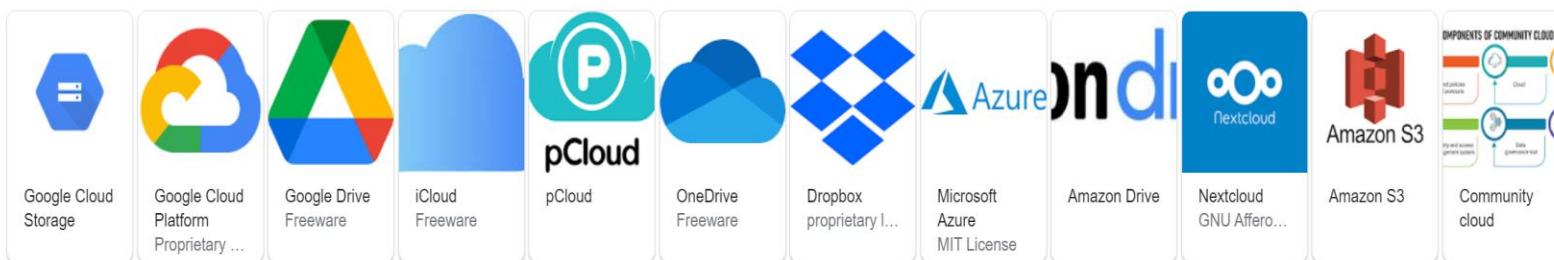
Drawbacks of Server Computing

Cost. High cost is probably the largest disadvantage of on-premise servers. First of all, the installation and upkeep of hardware and infrastructure, as well as cooling and power consumption, require significant investment. Secondly, you will need to hire an in-house IT staff to maintain and control the servers and ensure they properly function.

When to Use Server Computing?

Server computing is a valid option for companies that cannot entirely rely on a stable internet connection. Businesses with a well-skilled IT department ready to maintain servers, upgrade hardware, and solve issues themselves can also benefit from the on-premise server architecture.

Types of cloud storage



What is Serverless Computing?

Despite the name, serverless does not mean the absence of servers. Serverless architecture refers to delegating server management and maintenance to a third-party cloud provider. Companies can focus on developing products and writing code while the cloud vendor takes care of the server infrastructure. Let's investigate the reasons behind the rising adoption of serverless computing and look at its drawbacks.

Benefits and Disadvantages of Serverless Computing

Benefits of Serverless Computing

Cost. Serverless computing costs are not fixed and fluctuate depending on the needs of your company. As a result, you pay only for what you use and save a good portion of your budget. There is also no need to spend money on expensive hardware, paying high electricity bills, and keeping an IT team on staff to take care of the servers.

Scalability. Unlike traditional server computing, the majority of serverless providers offer their services on-demand. When you start out, and your application only requires minimal space and control, the cloud provider will provide a low-maintenance tier. However, as you grow and expand and require more power to handle thousands of users, the infrastructure will immediately match the demand.

Global data access. Your employees will be able to access files from anywhere and at any time. Since your cloud provider manages the servers, they are accessible from any device. This option is ideal for companies with remote workers and freelancers who can access essential data from any place in the world.

Table 6-1. Server vs Serverless

SERVER VS. SERVERLESS	SERVER COMPUTING	SERVERLESS COMPUTING
Security: It's a tie	Very secure option. Servers are stored on-site and are not accessible by non-authorized personnel. However, building security breaches can lead to data exposure and losses.	Servers are controlled by cloud vendors, making the data more susceptible to potential data breaches. However, data encryption and other security measures help keep data safe.
Cost: Serverless wins	Costs for on-premise servers are high as they include hardware installation and maintenance, expensive electricity bills, cooling tools, extra office space, additional human resources, etc.	Serverless architecture cuts the costs by 70% to 90% as you do not have to pay for extra services you do not require. Besides, no equipment, hardware, and HR costs add up to tremendous cost savings for you.
Control: Server wins	Having physical servers on the premises and managing all the operations in-house helps companies to preserve full control and have great visibility over processes.	Going serverless implies handing over the control to a cloud provider, thus losing some supervision over the operations.
Support: Serverless wins	Handling server-related issues in-house, you will have to strongly rely on your IT team. If they cannot solve a certain problem, you will lose access to your data for a long period of time, which will disrupt the business.	The cloud provider's responsibility is to handle the operations, ensure that everything works smoothly, and fix problems. They also have more experience solving such problems and will mostly figure it out in no time.
Scalability: Serverless wins	Resource scaling on-premise is expensive, slow, and difficult to execute. The time and money it takes to purchase, install, and set up the new equipment	Serverless computing is known for offering scalable solutions and providing on-demand resources. You can scale up and down almost instantly when

	puts server computing at a disadvantage.	something changes.
Accessibility: Server wins	As servers are on-site with you, you do not need an internet connection to access data. This is a great advantage for companies that reside in locations with unstable internet.	If you lose internet connection, you and your employees will not have any access to the data, which can damage the business.

Lack of control and visibility. Serverless is about delegating server maintenance, which subsequently gives away some control. This is why it is crucial to vet the companies and partner with a reliable and experienced cloud provider. You will likely lose some transparency either way, but you do not have to worry about data safety with a trustworthy partner.

When to Use Serverless Computing?

One of the biggest benefits of serverless computing is its scalability. If you're expecting growth and require an environment that will meet the demand and adapt to your needs, serverless is the right path.

Server vs. Serverless: Side-by-side Comparison

In this section, we will closely compare server vs. serverless to find out which option is the best for your case.

We choose to create server in local computer (high-performance computer) to put in it all database of students and professors and do all operation that we need through it ,we don't choose the serverless because it need high cost with less space compare to the local server can provide it

why we choose backend framework

Motivation: Why should you choose the best backend framework?

- You are a part of a startup, software house or what-do-they-call-us-these-days. Every IT company will eventually face this problem: engineers with different programming languages or framework backgrounds. You land a new project to build and you get to choose the technology. It requires two backend developers. Let us see... Who is free/bookable right

now... Great, one PHP developer and one Python developer. What should you do? You will probably evaluate who is more experienced and choose his main programming language as the implementation language. Or maybe you decide to start with one programmer and wait for another project to finish so that you have two people with the same background? What if you had a cyclic framework choosing process? You wouldn't have the problem. Benefits:

- Every backend developer would work on the same framework during some period of time.

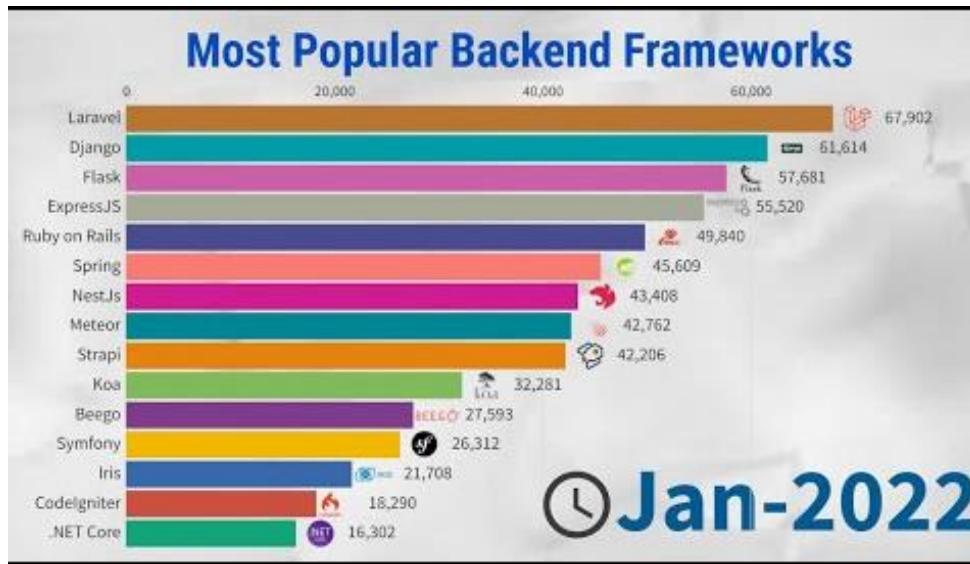


Figure 6-9. Popular Backend frameworks

- Estimations would be easier.
- Quality wouldn't suffer on the project start.
- Reviews would take less time.
- Products could be delivered faster.

Best Backend Frameworks

1. Python – Django, Flask
2. JavaScript – Express.js, Spring
3. PHP – Laravel, CakePHP

4. Ruby – Ruby on Rails
5. Elixir – Phoenix

6.9 Web Application Framework

What is a Web Application Framework?

- A web app framework is a software tool that is intended to facilitate and support web application development which includes web resources, services, and web APIs.
- They offer a specific way to form and install web applications on the **World Wide Web**.
- The frameworks aspire to mechanize the overhead linked with general activities performed in the development of the web.
- For instance, they supply libraries for templating, database access, and session management; also they frequently endorse code reuse.
- Even though they are aimed at the development of vibrant websites, they are appropriate for some static websites too.

Key Features

- The best frameworks for web development generally place the control flow of the program and enable the users to ‘hook into’ that flow by divulging in diverse events.
- Such an ‘**inversion of control**’ pattern is meant to be a crucial principle of a framework and benefits the coding by implementing a universal flow for the team to customize in similar conducts.
- After knowing what is a web framework is, let us examine some critical features that aid experienced developers in web app development.

1. Web template system:

This is a collection of diverse software and methodologies reinforced to form and deploy web pages. The template engines are utilized to execute these web templates. They are termed as a tool for web publishing.

2. Modern Applications:

Modern web applications are sustained by [robust web development tools](#). The most universal and one of the best frameworks for building applications facilitates the creation of all-purpose websites, forums, blogs, content management systems, to name a few.

3. Web Scaffolding:

This is an important system to understand and employ and is affiliated with MVC architecture. *The essential parts of any application or the complete project configuration can be produced by the framework mechanically.* This approach boosts the speed of the development cycle and regulates the codebase.

4. Web Caching:

It aids to store various documents and evades the infuriating occurrence of server overload. The users can exercise it in different systems if a number of conditions are satisfied.

5. High Security:

This security feature has ample criteria for recognizing and sanctioning or rejecting access to dissimilar functions. It also aids in identifying the profiles that utilize the applications to evade clickjacking. This makes the framework authorized and authentic.

What Are The Different Types of Web Frameworks?

The two primary functions of a website are on the server-side or client-side.

Server-side

The architecture of a server-side web framework enables the creation of landings, simple pages, and forms of different kinds. They can render output data and enhance security at the time of web attacks.

Such a framework functions mainly on meticulous and significant details, without which the application cannot run work effectively. All these vital traits simplify the backend web development procedure.

Client-side

With a client-side framework, you can easily enhance and execute a new user interface. It considers the work that takes place in the browser and has nothing to do with business logic.

Several animated characteristics can be fabricated with the frontend development frameworks and SPA ([single-page applications](#)).

Each client-side framework is different from the others in terms of its usage and functions.

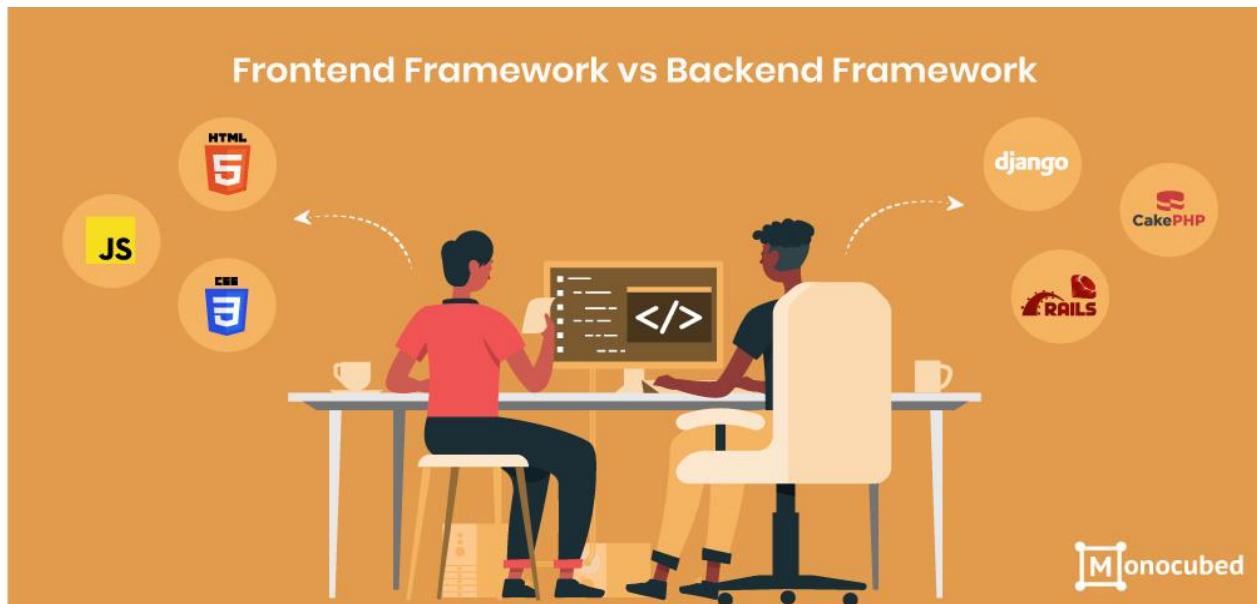


Figure 6-10. frontend & backend frame works

different choices for backend framework

Framework	Programming Language	Famous Use Cases
Django	Python	Instagram Pinterest Coursera
Laravel	PHP	Deltanet Travel Neighborhood Lender MyRank
Ruby on Rails	Ruby	ZendDesk Shopify GitHub
ExpressJS	NodeJS	MySpace GeekList Storify
CakePHP	PHP	Mapme Educationunlimited Followmy Tv
Flask	Python	Red Hat Rackspace Reddit
Asp .NET	C#	Microsoft Godaddy Ancestry
Spring Boot	Java	Trivago Via Varejo Intuit
Koa	NodeJS	—

Phoenix	Elixir	Financial Times Fox 10 ABC15
---------	--------	------------------------------------

6.10 Conclusion

At its core, a database system is a computerized record-keeping system; it stores and provides access to information. Reduced to its basic components, a database system consists of data, hardware, and software. Although these simple components remain fundamental to database systems, their scope, magnitude, and complexity have expanded mightily. The database research community plays a foundational role in creating the technological infrastructure from which database advancements evolve, We choose to create server in local computer (high-performance computer) to put in it all database of students and professors and do all operation that we need through it ,we don't choose the serverless because it need high cost with less space compare to the local server can provide it.

7. Chapter Seven: Mobile signal detector.

Mobile signal detector.

7.1 Mobile signal

How do cell phones work?

A cell phone does not use wires to transfer your voice. When you speak into a cell phone a microphone turns your voice into electrical signals. A microchip in the phone modulates (or varies) a **radio wave** using the electrical signal. The radio wave travels through the air to a nearby cell tower; the tower sends your voice to the person you are calling, and the process is reversed so that the person on the other end can hear your voice

A cell phone typically transmits and receives signals in the frequency range of 0.9 to 3GHz

Radio Waves

It's a wave from the portion of the electromagnetic spectrum at lower frequencies than microwaves.

The wavelengths is the longest of radio waves range from thousands of meters to 30 cm.

These correspond to frequencies as low as 3 Hz and as high as 1 gigahertz (10^9 Hz).

Radio-wave communications signals travel through the [air](#) in a straight line, reflect off of clouds or layers of the [ionosphere](#), or are [relayed](#) by [satellites](#) in space.

They are used in standard broadcast [radio](#) and [television](#), [shortwave radio](#), navigation and [air-traffic control](#), [cellular telephony](#), and even remote-controlled toys.

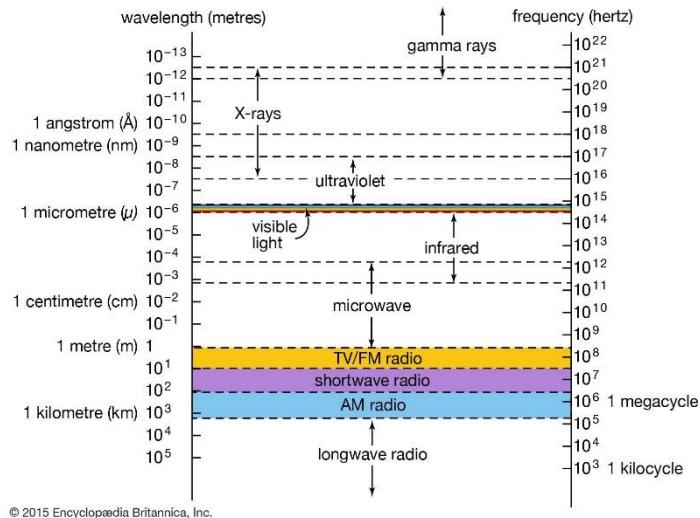


Figure 7-1. waves

Because electromagnetic radiation travels in free space in straight lines, late 19th-century scientists questioned the efforts of the Italian physicist and inventor [Guglielmo Marconi](#) to develop long-range radio.

Earth's curvature limits the line-of-sight distance from the top of a 100-metre (330-foot) tower to about 30 km (19 miles). Marconi's unexpected success in transmitting messages over more than 2,000 km (1,200 miles) led to the discovery of the [Kennelly-Heaviside layer](#), more commonly known as the [ionosphere](#). This

region is an approximately 300-km- (190-mile-) thick layer starting about 100 km (60 miles) above Earth's surface in which the atmosphere is partially ionized by [ultraviolet light](#) from the [Sun](#), giving rise to enough electrons and ions to affect radio waves.

electromagnetic waves, radio waves in a vacuum travel at the [speed of light](#), and in the Earth's atmosphere at a close, but slightly lower speed.

Radio waves are generated artificially by an electronic device called a [transmitter](#), which is connected to an [antenna](#) which radiates the waves. They are received by another [antenna](#) connected to a [radio receiver](#), which processes the received signal. Radio waves are very widely used in modern technology for fixed and mobile [radio communication](#), [broadcasting](#), [radar](#) and [radio navigation](#) systems, [communications satellites](#), [wireless computer networks](#) and many other applications. Different frequencies of radio waves have different propagation characteristics in the Earth's atmosphere; long waves can [diffract](#) around obstacles like mountains and follow the contour of the earth ([ground waves](#)), shorter waves can reflect off the [ionosphere](#) and return to earth beyond the horizon ([skywaves](#)), while much shorter wavelengths bend or diffract very little and travel on a [line of sight](#), so their propagation distances are limited to the visual horizon.

Communication

Radio waves transmitted by antennas in certain directions are bent or even reflected back to Earth by the ionosphere.

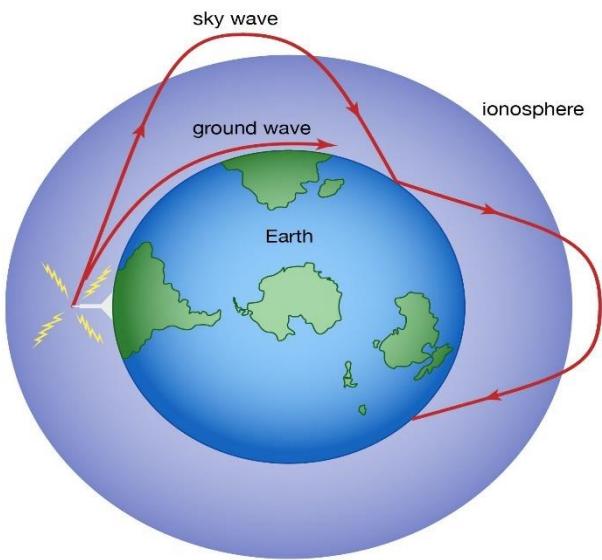


Figure 7-2. Ionosphere reflection.

They may bounce off Earth and be reflected by the ionosphere repeatedly, making radio transmission around the globe possible. Long-distance communication is further facilitated by the so-called ground wave. This form of electromagnetic wave closely follows Earth's surface, particularly over water, as a result of the wave's interaction with the terrestrial surface. The range of the ground wave (up to 1,600 km [1,000 miles]).

The bending and reflection of the sky wave by the ionosphere depend on the frequency of the waves.

Under normal ionospheric conditions 40 MHz is the highest-frequency radio wave that can be reflected from the ionosphere.

In order to accommodate a big band width signals, television frequencies are necessarily higher than 40 MHz.



Figure 7-3. Traditional communication

As a radio wave travels from the transmitting to the receiving [antenna](#), it may be disturbed by reflections from buildings and other large obstacles.

Disturbances arise when several such reflected parts of the wave reach the receiving antenna and interfere with the reception of the wave.

Radio waves can penetrate nonconducting materials, such as [wood](#), [bricks](#), and [concrete](#), fairly well.

They cannot pass through electrical [conductors](#), such as water or metals.

Above $v = 40$ MHz, radio waves from deep space can penetrate Earth's atmosphere.

This makes [radioastronomy](#) observations with ground-based telescopes possible

7.2 Detector is valuable

Implementation

Mobile phone detector is important to industry due to its ability to add safety to many businesses, offices, classrooms, etc.

For prison for example, individuals smuggle phones into prisons on a day-to-day basis, due to inmates using the mobile device to orchestrate an escape plan, orchestrating ways to smuggle more contraband into the prison, and much more that is not good for the prison system.

On top of that, individuals use mobile devices in classrooms to cheat on example and to send the answer around, so many schools implemented mobile devices to restrict cheating from occurring.

Also, for offices it can restrict distraction so that employees can focus the importance of each meeting he/she attends.

Lastly, within **classrooms** mobile devices can be a huge distraction from a student receiving his or her education.



7.3 Signal detector

Signal detectors concept

The [cell phone detector](#) is a technological device that detects the presence and existence of cell phones in an area or within a stipulated range of operation by detecting **radio waves**.

Once this cell phone detector detects **RF** , the phone detector system raises the alarm ([LED](#)) and speak to the mobile phone user to switch it off.

This is a great way to prevent the use of cell phones in exams and lectures classrooms.



Figure 7-4. LED

How to build a detector? First, while detector depends on detecting the radio wave so we must have an antenna, which can receive these signals.

Second, this signal is weak .. needs an operational amplifier to amplify it.

Third, an indicator for this signal as a LED or buzzer.

Fourth, resistors and capacitors with specific values.

Fifth, Inductors if needed.

Sixth, a transistor or diode

The type of detector depends on the type of antenna

Types of antennas

- Dipole antenna
- Yagi antenna
- Monopole antenna
- Helix antenna
- Loop antenna
- Array antenna
- Microstrip antenna

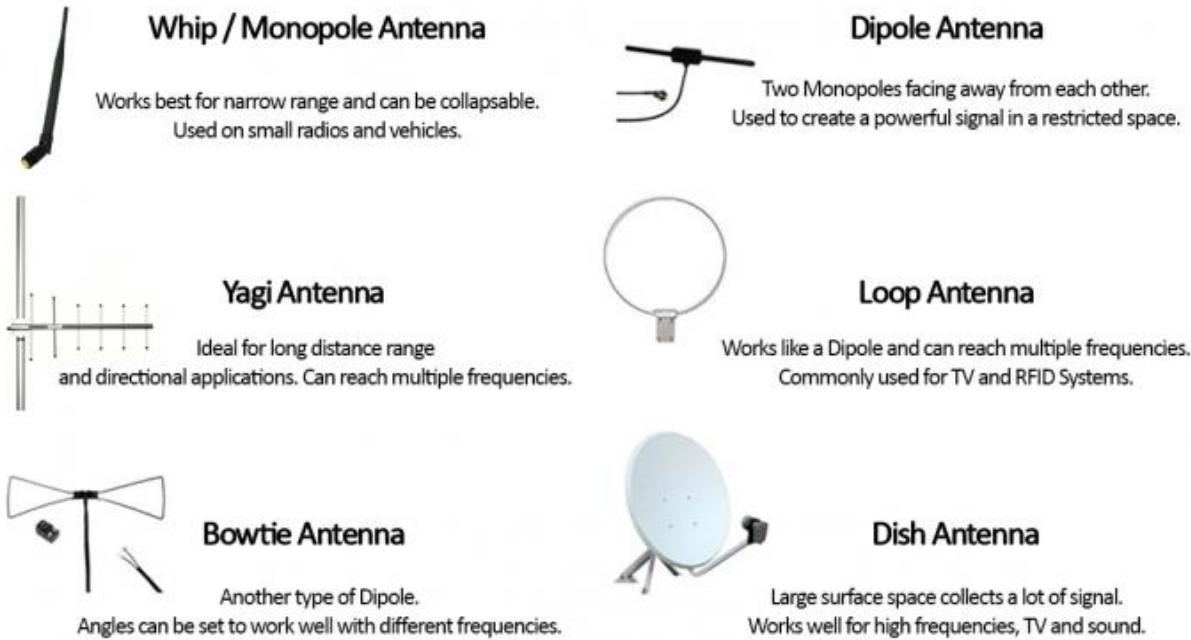


Figure 7-5. Types of antenna

Types of detectors?

Detectors names is taken from their components.

Such as,

1-Schottky diode circuit

This circuit have a unique property of being able to rectify low frequency signals, with low noise rate.

When an inductor is place near the RF signal source, it receives the signal through mutual induction.

This signal is rectified by the Schottky diode.

This low power signal can be amplified and used to power any indicator like an LED.

The Op-amp part of the circuit acts as the RF Signal Detector while Transistor part of the circuit act as the indicator.

The capacitors collection along with the antenna are used to detect RF Signals when a cell phone makes (or receives) a phone call or sends (or receives) a text message.

The detector circuit consists of an inductor, diode, a capacitor and a resistor. Here an inductor value of 10uH is chosen. A Schottky diode BAT54 is chosen as the detector diode, which can rectify low frequency AC signal. The filter capacitor chosen in a 100nF ceramic capacitor, used to filter out AC ripples. A load resistor of 100 Ohms is used.

Components required :

V1 = 12V

L1 = 10uH

R1 = 100Ohms

C1 = 100nF

R2 = 100K

R3 = 3K

Q1 = BC547

R4 = 200 Ohms

R5 = 100 Ohms

IC1= LM339

R6 = 10 Ohms

LED = Blue LED

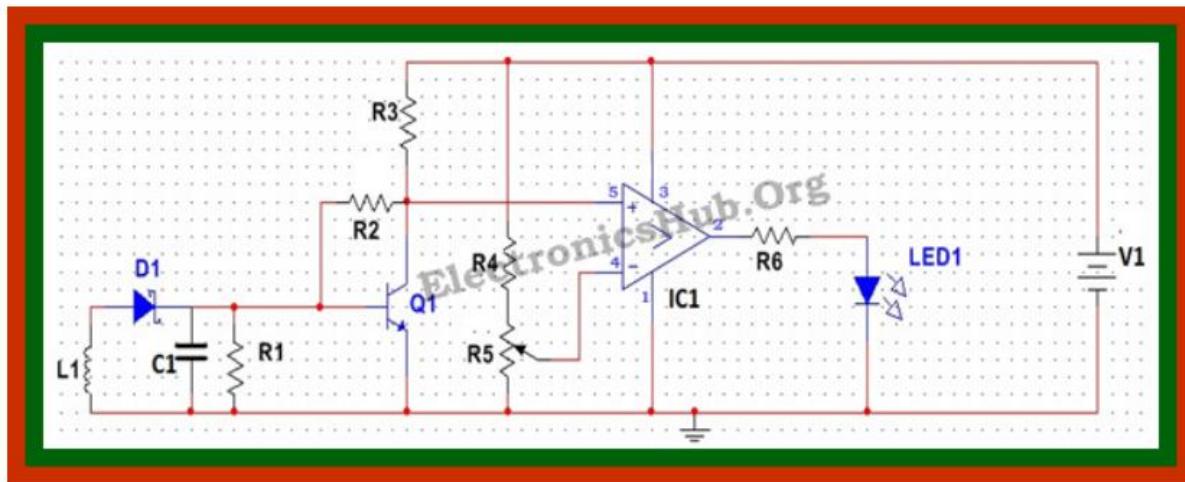


Figure 7-6. Simple cell phone detector circuit

2-Simple cell phone detector circuit

The Op-amp part of the circuit acts as the RF Signal Detector while Transistor part of the circuit act as the indicator. The capacitors collection along with the antenna are used to detect RF Signals when a cell phone makes (or receives) a phone call or sends (or receives) a text message.

Op-Amp reads the signals by converting the rise in current at input to voltage at output and the LED will be activated.

Components required

- CA3130 Op-Amp
- Resistors – $2.2M\Omega \times 2$, $100K\Omega$, $1K\Omega$
- Capacitors – $22pF \times 2$, $0.22nF$, $47pF$, $100\mu F$
- BC548 NPN Transistor
- LED
- **Antenna**
- Connecting Wires
- Breadboard
- 9V Battery

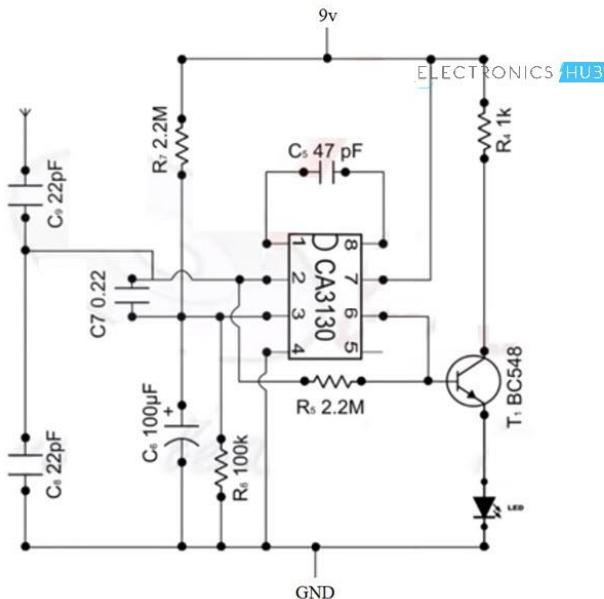


Figure 7-7. Simple cell phone detector circuit 2

7.4 (Feedback/Capacitor) Detector

Principle of circuit

Main idea here is the signal power is stored in the chemical capacitor through the wire.

The resistors and ceramic capacitors connection acts as a filter to select the desired signal (the RF frequency at which the phone operates).

The resistors and capacitor value are chosen such that the particular frequency is detected (**RF frequency**).

The operational amplifier amplifies the captured RF signal because it is too low and a proper signal level needs to be applied to the base of the transistor for it to turn ON.

So, the output signal from the OPAMP with a 0.7 V drop (**The forward voltage dropped across the diode in the transistor**) forward biases the LED and turns it ON.

“ It based on noise signals and feedback it to amplified to reach steady state case(the operating resonance frequency). “

Call Detector Circuit are useful electronic circuits than can detect the presence of any transmitting RF mobile device nearby and gives a visual indication with respect to it. Basically, Cell-phone detector is a Frequency Detector or a Current to Voltage Converter Circuit. With the development of communication technology, the requirement of cellphones has expanded dramatically.

A cellphone generally transmits and receives indicators in the frequency range of **0.9 to 3GHz**. So today in this tutorial, we will go through step by step instructions on “How to make a simple **Mobile Phone Detector** circuit” using a CAC3130 Operational amplifier

The working of this circuit is pretty simple.

A 0.22uF capacitor captures the RF signal from any transmitting device namely a mobile phone. The capacitor leads act as a small gigahertz loop antenna to capture the transmitting RF signals from a mobile phone.

CA3130 functions as a current to voltage converter with a 0.22uF capacitor connected between its inverting and non-inverting input channels. When the cell phone detector signal is detected, the output of IC becomes high and low alternately according to the frequency of the signal as indicated by the LED.



Figure 7-8. CA313 IC (OP-AMP)

Datasheet

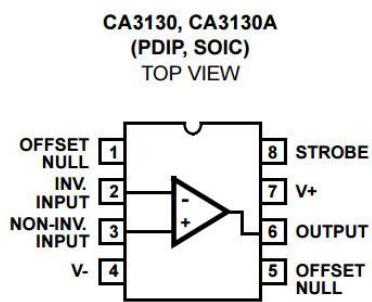


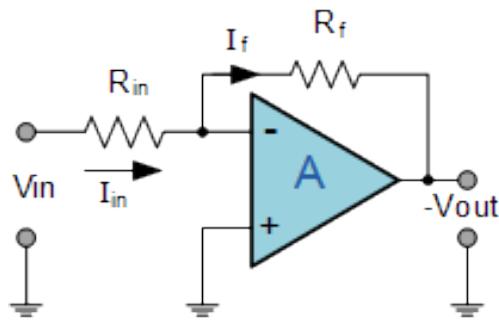
Figure 7-9. Data sheet of CA313 IC (OP-AMP)

The pin configuration of CA3130 CMOS Op-Amp is shown below. This IC includes 8-pins where each pin & its functionality is discussed below.

- Pin1 (Offset Null Pin): It is used to fix the offset voltage
- Pin2 (Inverting Pin IN –): A stable voltage is given to this IN-pin
- Pin3 (Non-inverting Pin IN +): An uneven voltage is given to this IN+ pin
- Pin4 (Vcc-): It is a GND pin
- Pin5 (Offset Null Pin): It is used to fix the offset voltage
- Pin6 (Output): The o/p pin of the IC
- Pin7 (Vcc+): It is a voltage supply pin where the voltage ranges from 5V – 16V
- Pin8 (Strobe): It is used to switch off the o/p stage

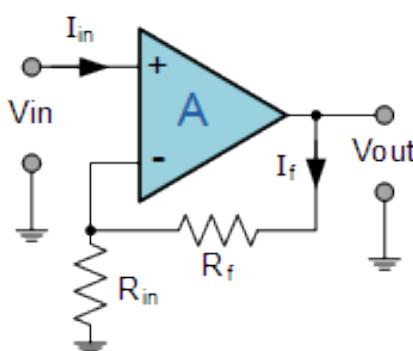
Working

Inverting Op-amp



$$A = \frac{V_{out}}{V_{in}} = -\frac{R_f}{R_{in}}$$

Non-inverting Op-amp



$$A = \frac{V_{out}}{V_{in}} = 1 + \frac{R_f}{R_{in}}$$

Table 7-1. Comparison between inverting Op-amp & Non-inverting Op-amp

At first, this IC works as a comparator between the two input pins.

Second, for With + voltage of (**9V**)

And - voltage of (**0V**).

If the input in positive pin is larger than negative pin, 80% positive voltage will be multiplied by Gain and go through output (**Vout**).

While feedback the out put to the negative pin again.

The **features and specifications of the CA3130 CMOS op-amp** include the following.

- Max sink current is 20mA
- i/p terminal current is 1mA
- Duration of o/p short circuit is Indefinite
- CMRR (Common Mode Rejection Ratio) is 80dB
- DC i/p voltage ranges from +8V to -0.5V
- The power supply range is wide
- Differential i/p voltage is 8 V
- Single supply ranges from – 5V to 16V
- DC voltage supply is 16 V
- Double supply ranges from – ±2.5V to ±8V
- Common mode i/p voltage range mainly includes -ve supply rail and input terminals can swing 0.5V under -ve supply rail
- Output Voltage Max is 13.3V
- CMOS o/p stage allows swing of signal to any or both supply rails
- Op-amp included with MOSFET on the output
- It is perfect for the applications of a single supply
- Max source current is 22mA
- Current supply is 10mA

This BiMOS op-amp incorporates MOSFET, so it is applicable in oscillator circuits, mobile jammers & frequency generators. The input terminal current used by this IC is approximately 1mA & the maximum o/p voltage is 13.3V.

The CMRR of CA3130 IC is 80dB. Here, the rejection ratio is the capacity of the device to refuse common-mode signals.

So it is the ratio of differential-mode gain & common-mode gain.

The CA3130 is available with high i/p impedance because of the in-built MOSFETs. Whenever the sensor output voltage is connected to the inverting and non-inverting terminals of the IC, then the sensors output voltage doesn't modify its value.

If you are looking for an IC with low power utilization, high BW, quick sample rate, max i/p impedance, and then this IC is suitable for you.

How to use CA3130 CMOS Op-Amp?

CA3130 CMOS op-amp performs single & dual supply functions but the main difference between them is, the 4th pin is connected to GND in single-supply operation whereas the pin-8 strobes the o/p phase into quiescence.

This op-amp is used for compensation of phase to drive CMOS digital circuits within the applications of the comparator. Offset nulling is attained by connecting a potentiometer across pins 1 & 5.

The potentiometer's slider arm is allied to pin 4 so that a fine value can be attained through sliding the arm. For obtaining effect compensation, a tiny capacitor is used in between pins 1 & 8. So for providing sufficient compensation, a 47pF capacitor is suitable.

Transistor BC-548

Datasheet

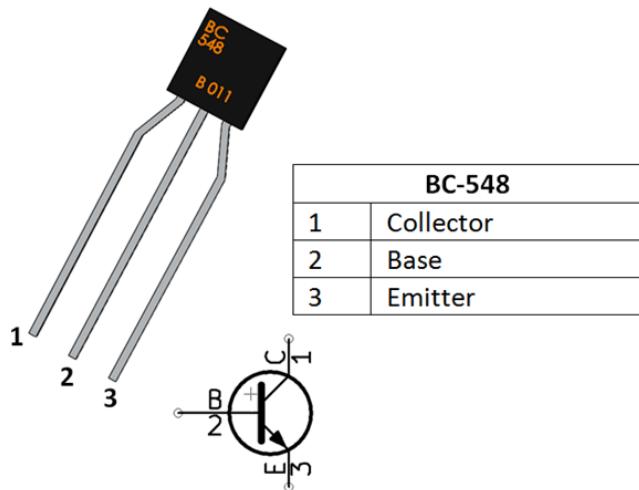


Figure 7-10. Transistor BC-548 Datasheet.

Resistors



Figure 7-11. Two $2.2\text{ M}\Omega$ Resistors

- $1\text{ K}\Omega$



Figure 7-12. 1K Resistor

- $100 \text{ K}\Omega$



Figure 7-13. $100 \text{ K}\Omega$ Resistor.

Capacitors



Figure 7-14. 47 pF Capacitor

- 47 pF
- $0.22 \mu\text{F}$

This capacitor is largely **important..**



Figure 7-15. $0.22 \mu\text{F}$ Capacitor

1. The capacitor starts being charged until the blue LED is off.
 2. A constant voltage is dropped in pin 3 and 2.
 3. Capacitor receives **RF** through its leads
 4. Capacitor starts charging and discharging according to the received sinusoidal wave.
 5. At positive peaks the capacitor produces a current through its wires which led to output through IC.
 6. The output is amplified (multiplied by G).
- 100 uF (Chemical)

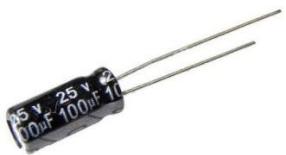


Figure 7-16. 100 uF (Chemical)

- Anti LED ripple capacitor (optional output smoothing)
0.47 uF (Chemical)



Figure 7-17. 0.47 uF (Chemical)

LED



Purpose of each component

Capacitor is used to store energy received by wires and feed it back to the IC many times until reaching the steady state case before the IC amplifies the input noisy signal.

IC CA3130 contains the advantages of **CMOS** and **bipolar**.

Resistors with specific values are determined to control the current throughout the circuit.

Transistor is used to switch the electronic signals and electrical power.

Light-emitting diode is used to provide light for the final output.

Power Supply is used to provide the circuit with a proper amount of voltage.

The digital multimeter is used to set the power supply to the required voltage amount.

Each **capacitor** is used to store energy and provide the circuit with energy when it is necessary.

Procedure

1. Connect the digital multimeter and power supply together.
2. Set the power supply to 9 volts.
3. Place a wire for voltage, as well as, a wire for the ground.
4. Place CA3130 IC.
5. Connect pin 7 of the CA3130 IC to the voltage source.
6. Connect pin 4 of the CA3130 IC to the ground.
7. Place the BC548 transistor.
8. Connect Pin 6 of the CA3130 IC to the base of the BC548 transistor.
9. Connect the collector of the BC548 transistor to the voltage source using the $1\text{k}\Omega$ resistor.
10. Connect the anode of the light-emitting diode to the emitter of the BC548 transistor and the cathode of the light-emitting diode is connected to the ground.
11. Connect pin 2 of the CA3130 IC to the base of the BC548 transistor using a $2.2\text{M}\Omega$ resistor.
12. Connect pin 3 of the CA3130 IC to the ground using a $100\text{k}\Omega$ resistor.
13. Place the $100\mu\text{F}$ capacitor.
14. Connect the anode of the $100\mu\text{F}$ capacitor to pin 3 of the CA3130 IC, as well as, the cathode of the $100\mu\text{F}$ capacitor to the ground.
15. Connect pin 3 of the CA3130 IC to the voltage using a $2.2\text{M}\Omega$ resistor.
16. Connect pin 2 to pin 3 of the CA3130 IC using a $0.22\mu\text{F}$ capacitor.

17. Connect pin 1 to pin 8 of the CA3130 IC using a 47pF capacitor.

Circuit Diagram

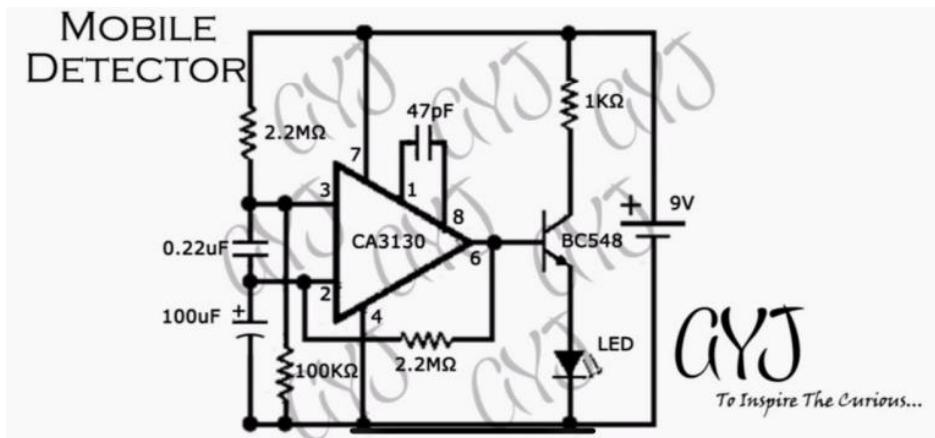


Figure 7-18. Circuit Diagram of mobile detector.

Circuit on Breadboard

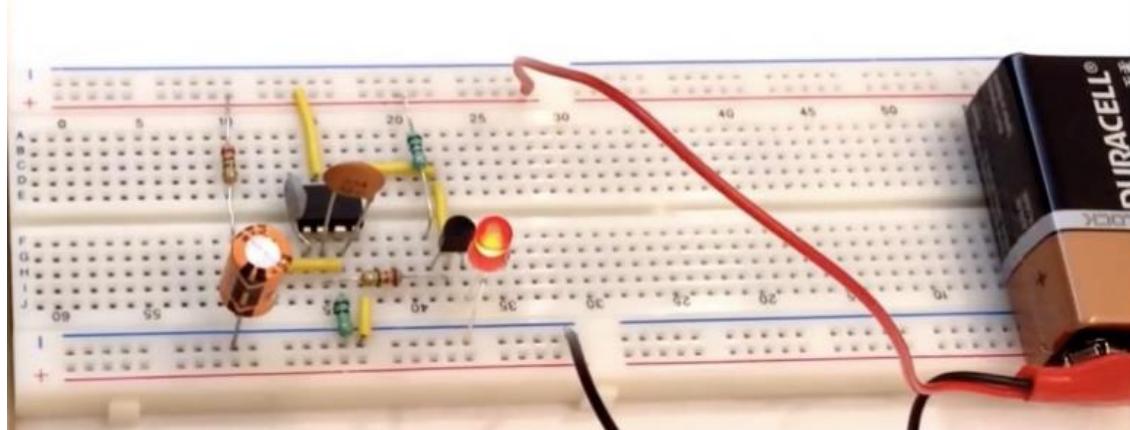


Figure 7-19. Circuit on test Breadboard

Schematic diagram

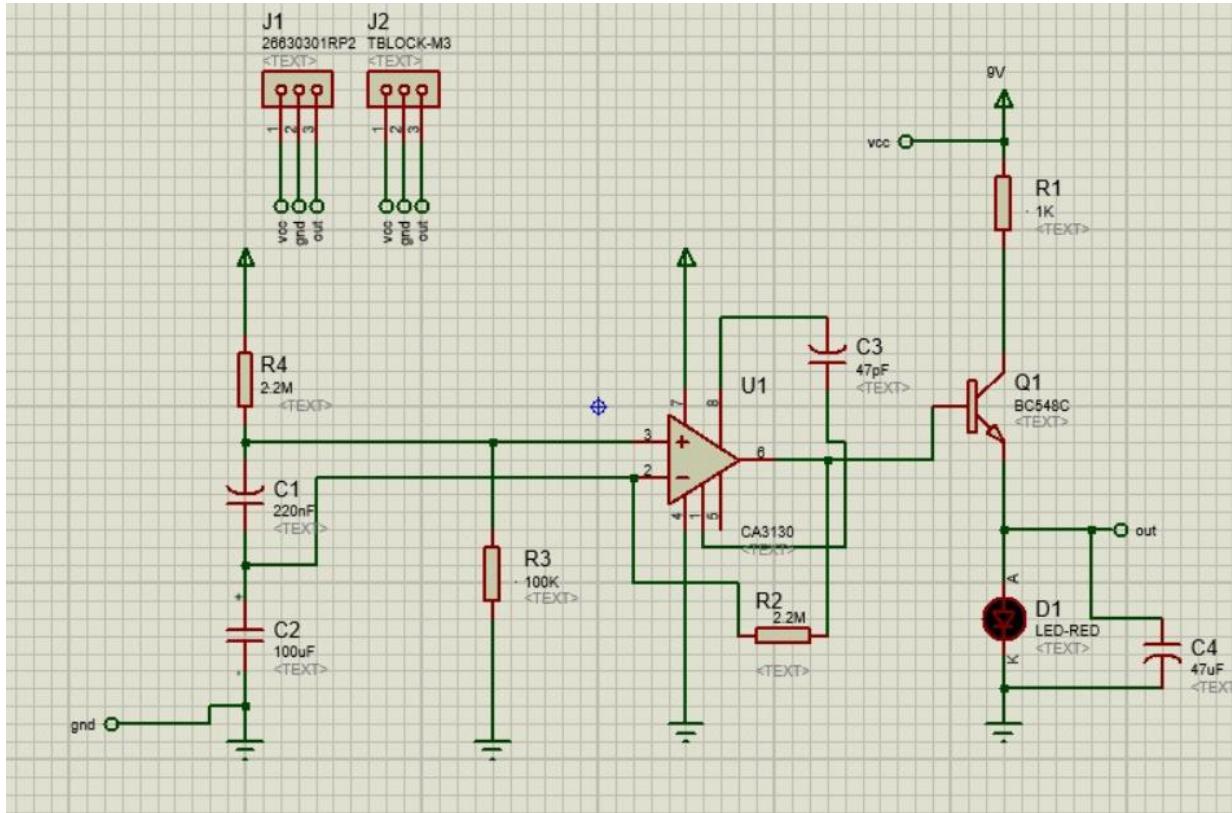


Figure 7-21. Schematic diagram

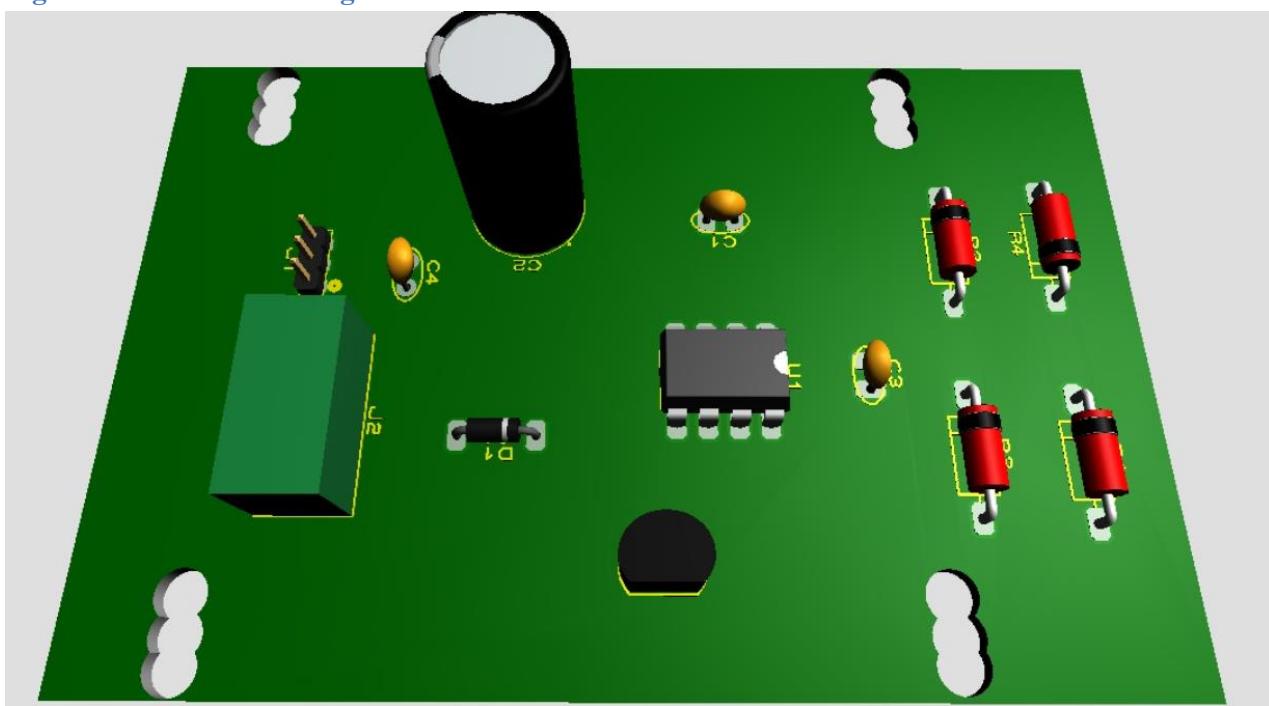
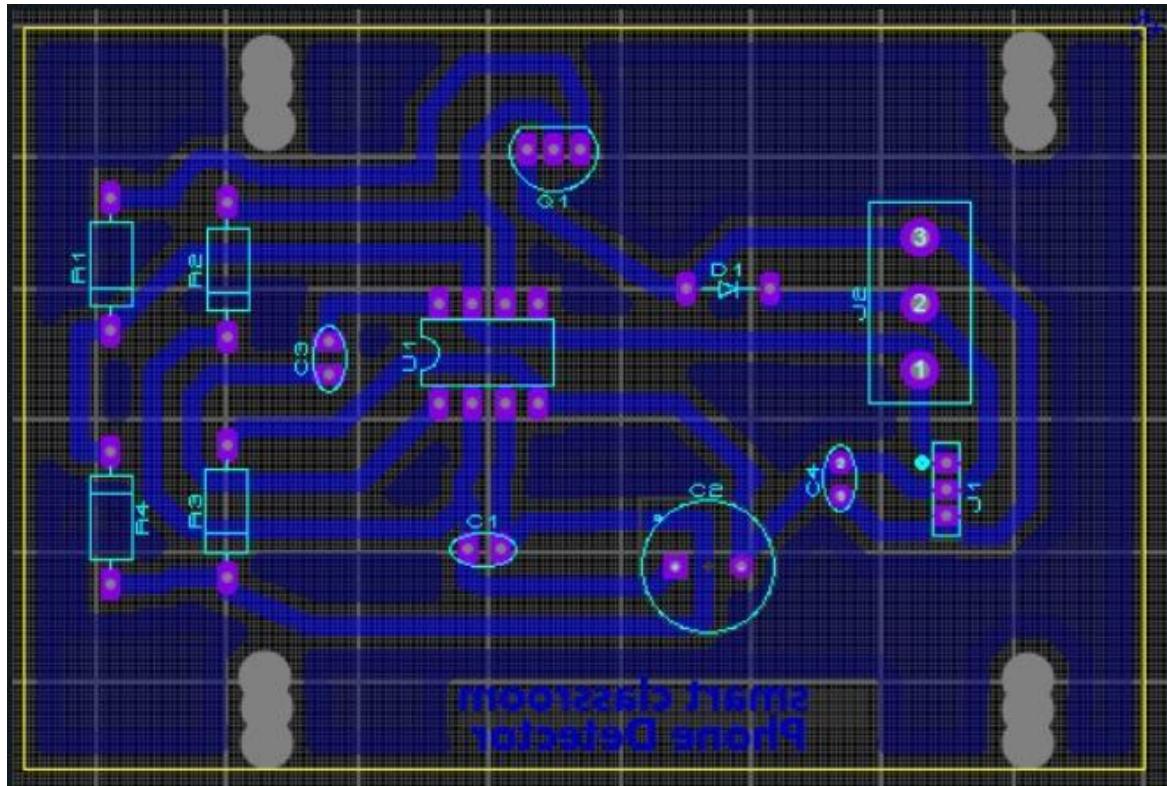
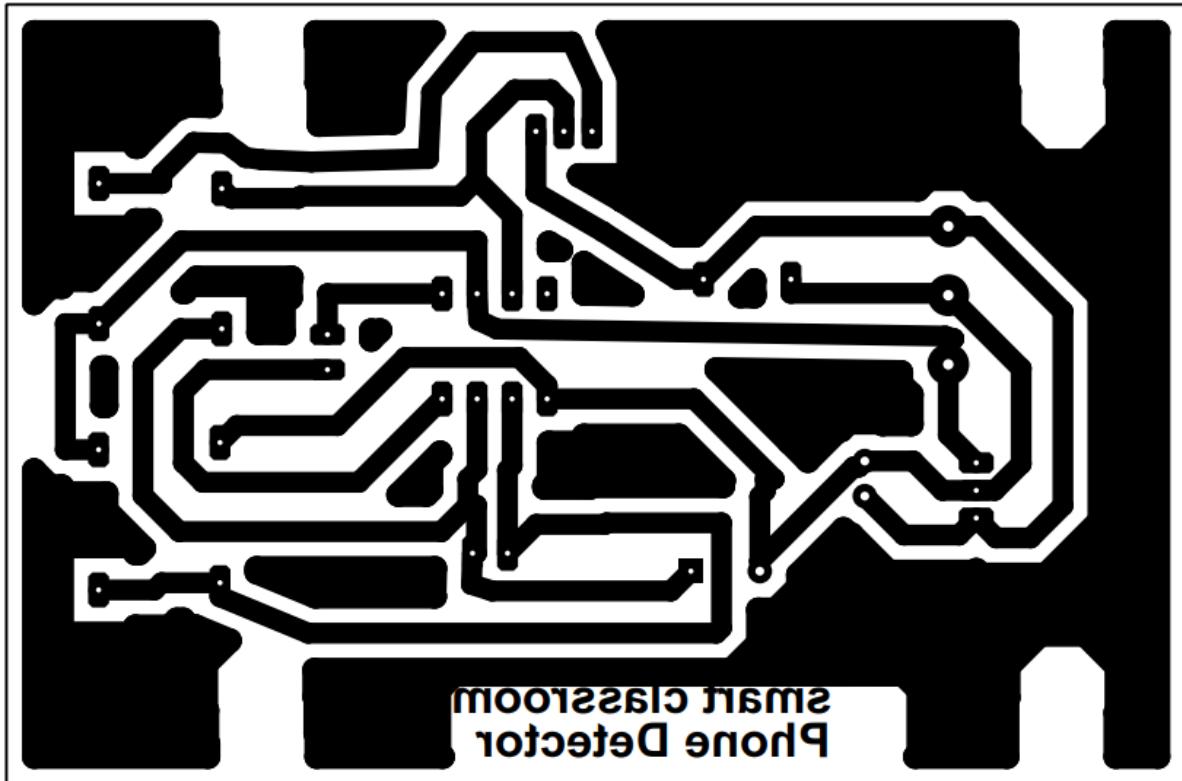


Figure 7-20. Peb circuit

Circuit Layout



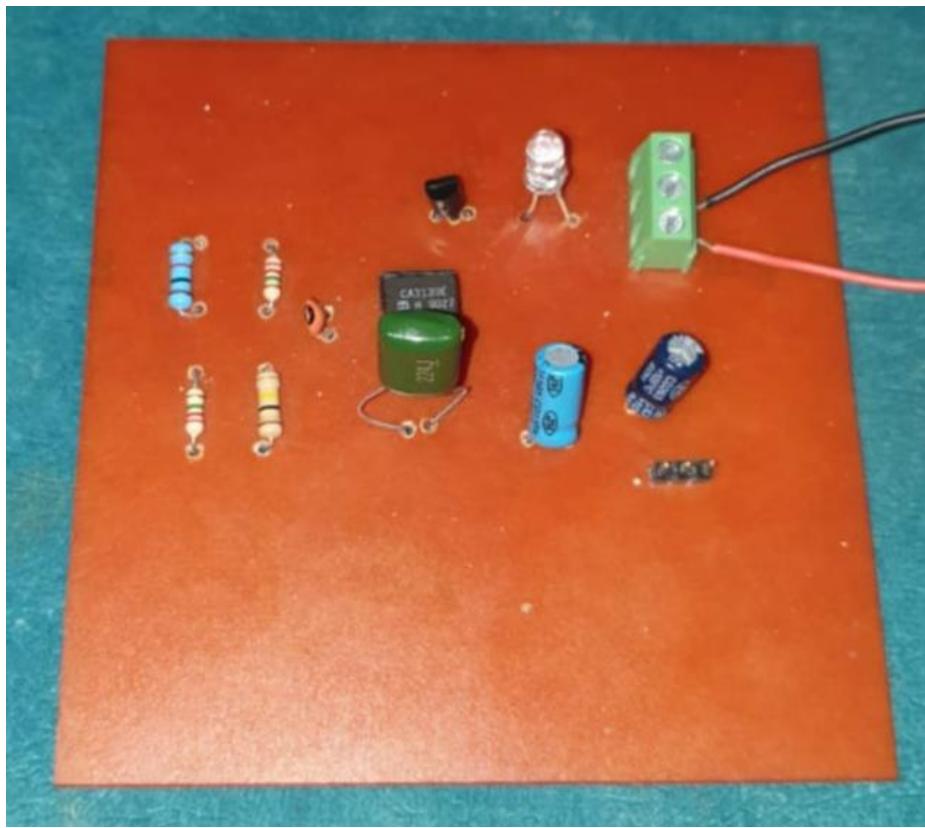


Figure 7-22. Circuit on PCB

Cost

Table 7-2. Component and its Cost

Components	Count	value	Price
CA3130	1	-	17.00 LE
Resistors	4	2.2MΩ-1KΩ-100KΩ	5.00 LE
Capacitors	3	47pF-0.22uF-100uF	5.00 LE
Transistor BC548	1	-	15.00 LE
LED	1	-	1.00 LE
PCB	1	-	15.00 LE
9V Battery	1	-	15.00 LE
Etching Solution	1	125ml	20.00 LE
Metallic marker	1	-	50.00 LE
Total	-	-	143.00 LE

Conclusion and Future work

We presented Smart classroom is representative of modern teaching. With the advent of modern technology it becomes easier for the students as well as teachers to perform their task more efficiently. The smart classroom is rectification to various problems that teachers as well as students face inside the classroom. This project tended to make use of modern technology for helping the teachers in utilizing more time for teaching and students to easily get access to the study material. In this project we constructed a prototype of a smart classroom by adding some tools that help the students and the lecturer and reduce time wasting during classes like face detection and recognition and detect him in exams and to get attendance and voting and send this data to a local server and detect the try of cheating if any student tried to make a call during exam and other tools, this application would be pivotal for carrying out various operations in the classroom.

Our project is implemented in many stages like dealing with The Raspberry Pi with its powerful little beast and a great platform for building low-cost, but highly capable, embedded systems. The interfaces built into its GPIO connector make it easy to bolt on modules using simple low-cost electronics and a bit of configuration to create very functional and flexible systems. In our project we use this function for two reasons:

1. In lectures when the professor wants to take the absence of students, the camera of the raspberry pi open to check who is present.
2. Is there is an exam the camera raspberry pi open to check the required student is using the raspberry pi or there is another one? If there another one the camera check if the student is another student in the class or it's an unknown person.

We used Webscoket and PHP in order to communicate with the server and each one is used in its own application as we discussed in detail in Web application front end chapter, and when we had discussed our core features of Our Smart Classroom server and the internal implementation of its different components and the multiple tech stack and protocol that power them. And we used many web standards to achieve full-duplex bidirectional communication that is used in the smart classroom as we discussed in WebSocket chapter, the database and reviewed how a database system is a computerized record-keeping system; it stores and provides access to information. Reduced to its basic components, a database system consists of data, hardware, and

software. Although these simple components remain fundamental to database systems, their scope, magnitude, and complexity has expanded mightily we represented our project database physical model for our project and its implementation. Any module or platform like Raspberry Pi we used to enhance the classroom using this IoT environment as we see from our point of view is suitable for both teachers and students, this development technologies is not only for this usage or for this methodology or The development of the educational process there are a lot of benefits to gain using it. And because of that any technology has its own target we modified this technology to be suitable for our project. There good point that it is easy for our project to be suitable for development and modification or for any other application be modifying the coding of it. For example we used html it is very simple language for modifying it to any other purposes. From our point of view there is a lot of things to enhance and apply development for our project like focus on physical environment in the classroom. Also at case of preventing Cheating methodology can be improved by different technologies enhancement. Also at case of vote methodology can be improved by different technologies enhancement. The GUI of exam can be improved by adding other different questions types not only MCQ so there will be a variation in questions. There much more enhancement can be added to improve interface between the lecturer and his students. Webserver can be global on the internet not local server so the students can know about his case and his classes information from anywhere not only locally. Also Phone detector module can be improved.

References

1. Saini, Mukesh Kumar, and Neeraj Goel. "How smart are smart classrooms? A review of smart classroom technologies." *ACM Computing Surveys (CSUR)* 52.6 (2019): 1-28.
2. Cebrián, Gisela, Ramon Palau, and Jordi Mogas. "The smart classroom as a means to the development of ESD methodologies." *Sustainability* 12.7 (2020): 3010.
3. Priya Pasumarti, P. Purna Sekhar Classroom Attendance Using Face Detection and Raspberry-Pi.
4. Revathi, R., M. Suganya, and Gladiss Merlin NR. "IoT based Cloud Integrated Smart Classroom for smart and a sustainable Campus." *Procedia Computer Science* 172 (2020): 77-81.
5. Md. Sanwar Hossain, Mostafizur Rahman , Md. Tuhin Sarker Md. Ershadul Haque , and Abu Jahid, A Smart IoT Based System for Monitoring
6. M. Tissenbaum and J. D. Slotta, "Developing a smart classroom infrastructure to support real-time student collaboration and inquiry: a 4-year design study," *Instructional Science*, vol. 47, no. 1, pp. 1–40, 2019.
7. Petrović, Luka, et al. "Designing an extended smart classroom: An approach to game-based learning for IoT." *Computer applications in engineering education* 30.1 (2022): 117-132.
8. Srinivasan, C. R., et al. "A review on the different types of Internet of Things (IoT)." *Journal of Advanced Research in Dynamical and Control Systems* 11.1 (2019): 154-158.
9. Hossein Motlagh, Naser, et al. "Internet of Things (IoT) and the energy sector." *Energies* 13.2 (2020): 494.
10. T Kraska, M Alizadeh, A Beutel, EH Chi, J Ding... - 2021 Smart home.

11. P. W. Kim, "Ambient intelligence in a smart classroom for assessing students' engagement levels," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 10, pp. 3847–3852, 2019.

- 12.Kishore Kodali, Ravi & Yerroju, Subbachary (2018) “Energy Efficient Home Automation Using IoT.” *2018 International Conference on Communication, Computing and Internet of Things (IC3IoT)* : 151–154.
- 13.Mohammadi, Venus, Rahmani, Amir, Mohammed Darwesh, Aso, Sahafi, Amir. (2019) “Trust-based recommendation systems in Internet of Things: a systematic literature review.” *Human-centric Computing and Information Sciences* **9** (21): 1–61.
- 14.Alem olakovi and Mesud HadÅiali. (2018) “Internet of Things (IoT): A review of enabling technologies, challenges, and open research issues.” *Computer Networks* **144** (24): 17–39.
- 15.M. Elsaadany, A. Ali, W. Hamouda, ”Cellular LTE-A technologies for the future Internet-of-Things: Physical layer features and challenges”, IEEE Commun. Surveys Tuts., vol. 19, pp. 2544-2572, 4th Quart. 2017.
- 16.M. Elsaadany, A. Ali, W. Hamouda, ”Cellular LTE-A technologies for the future Internet-of-Things: Physical layer features and challenges”, IEEE Commun. Surveys Tuts., vol. 19, pp. 2544-2572, 4th Quart. 2017.
- 17.Y.-T. Lin, “Impacts of a flipped classroom with a smart learning diagnosis system on students’ learning performance, perception, and problem solving ability in a software engineering course,” *Computers in Human Behavior*, vol. 95, pp. 187–196, 2019.
- 18.J. Aguilar, J. Altamiranda, and F. Diaz, “Specification of a managing agent of emergent serious games for a smart classroom,” *IEEE Latin America Transactions*, vol. 18, no. 1, pp. 51–58, 2020.
- 19.Y. Huda and D. Faiza, “Desain sistem pembelajaran jarak jauh berbasis smart classroom menggunakan layanan live video webcasting,” *Jurnal Teknologi Informasi Dan Pendidikan*, vol. 12, no. 1, pp. 25–32, 2019.

20. Huang, Li-Shing, Jui-Yuan Su, and Tsang-Long Pao. "A context aware smart classroom architecture for smart campuses." *Applied Sciences* 9.9 (2019): 1837.
21. Saini, Mukesh Kumar, and Neeraj Goel. "How smart are smart classrooms? A review of smart classroom technologies." *ACM Computing Surveys (CSUR)* 52.6 (2019): 1-28.
22. Paul, Chinju, Amal Ganesh, and C. Sunitha. "An IoT-based smart classroom." *International Conference on Computer Networks and Communication Technologies*. Springer, Singapore, 2019.
23. Alfoudari, Aisha M., Christopher M. Durugbo, and Fairouz M. Aldhmour. "Understanding socio-technological challenges of smart classrooms using a systematic review." *Computers & Education* 173 (2021): 104282.
24. Jolles, Jolle W. "Broad-scale applications of the Raspberry Pi: A review and guide for biologists." *Methods in Ecology and Evolution* 12.9 (2021): 1562-1579.

25. Chaudhari, Ketulkumar Govindbhai. "Windmill Monitoring System Using Internet of Things with Raspberry Pi." *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering* 8.2 (2019): 482-485.

26. Balon, Branko, and Milenko Simić. "Using Raspberry Pi computers in education." 2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). IEEE, 2019.

27. Dhiman, Gaurav. "An innovative approach for face recognition using raspberry Pi." *Artificial Intelligence Evolution* (2020): 102-107.

28. Wang, Vanessa, Frank Salim, and Peter Moskovits. "The websocket protocol." *The Definitive Guide to HTML5 WebSocket*. Apress, Berkeley, CA, 2013. 33-60.

29. Saint-Andre, Peter. Extensible messaging and presence protocol (XMPP): Core. No. rfc6120. 2011.

- 30.Bray, Tim. The javascript object notation (json) data interchange format. No. rfc7159. 2014.
- 31.Zachmann, Gabriel. "OIDC-Agent: Managing OpenID Connect Tokens on the Command Line." SKILL 2018-Studierendenkonferenz Informatik (2018).

الملخص

مع ظهور المنازل الذكية والمدن الذكية وكل شيء ذكي ، بُرِزَ إنترنت الأشياء (IoT) كمجال ذو تأثير وإمكانيات ونمو مذهل ، حيث كانت تتوقع شركة Cisco Inc أن يكون لديها 50 مليار جهاز متصل بحلول عام 2020.

الإنترنت ، اختراع ثوري ، يتحول دائمًا إلى نوع جديد من الأجهزة والبرامج مما يجعله أمرًا لا مفر منه لأي شخص. ، لكن إنترنت الأشياء (IoT) يعد بمستقبل عظيم للإنترنت حيث نوع الاتصال هو آلة – آلة.

نقدم الفصول الذكية للتدرис الحديث. مع ظهور التكنولوجيا الحديثة ، أصبح من السهل على الطلاب والمعلمين أداء مهامهم بشكل أكثر كفاءة. الفصل الدراسي الذكي هو تصحيح لمختلف المشكلات التي يواجهها المعلمون والطلاب داخل الفصل. يهدف هذا المشروع إلى الاستفادة من التكنولوجيا الحديثة لمساعدة المعلمين في استغلال المزيد من الوقت للتدرис والطلاب للوصول بسهولة إلى المواد الدراسية. في هذا المشروع ، نقوم ببناء نموذج أولي للفصل الدراسي الذكي من خلال إضافة بعض الأدوات التي تساعد كلاً من الطالب والمعلم مثل سرعة تحديد وجه الطالب والتعرف عليه أثناء الامتحانات أو تحديد الطالب الحاضرين أو سرعة الحصول على نتائج الاستفتانات ورفع البيانات على الإنترت تلقائياً وتحديد تلقائياً حالات الغش عندما يكون هناك محاولة لإجراء اتصال خارجي أثناء فتره الامتحانات وكذلك بعض الأدوات الأخرى المتاحة للمحاضر لتوفير الجهد والوقت أثناء المحاضرة وبالتالي يكون التطبيق محوريًا لتنفيذ العمليات المختلفة في الفصل الدراسي.



جامعة بنها

كلية الهندسة بنها

قسم الهندسة الكهربائية

فصل ذكي باستخدام الراسبيري باي

تم تقديم تقرير التخرج عن المشروع وفاءً جزئياً للحصول على درجة البكالوريوس. إجازة في الهندسة الكهربائية ، كلية الهندسة بنها ، جامعة بنها

قدم بواسطه:

1. محمد محمد عبدالحافظ طنطاوي.
2. حسام اشرف عبد الفتاح احمد يوسف.
3. عبد الرحمن صابر عبد الفتاح مصطفى.
4. عبد الرحمن محمد جاد العزب.
5. محمد السيد عبدالسلام مبارك.
6. هشام عبدالله نور الدين محمد احمد.
7. محمود محمد جوده المتولى.

تحت اشراف:

د: ساره حمدي.

م: احمد رجب.

بنها - مصر

٢٠٢٢ يوليو