

PROJECT REPORT

On

LIVA SMART HEALTHCARE

Submitted to Rajasthan Technical University
in partial fulfillment of the requirement for the award of the degree of

B. Tech.

in

Computer Science and Engineering

Submitted By

Ishan Verma (PIET15CE044)

Akshita Rajoria (PIET15CE013)

Laveena Chaturvedi (PIET15CE054)

Under the Guidance of

Mr. Ajay Saini

at



**POORNIMA INSTITUTE OF ENGINEERING & TECHNOLOGY,
JAIPUR**

RAJASTHAN TECHNICAL UNIVERSITY, KOTA

APRIL, 2019

CERTIFICATE

This is to be certified that the project entitled “ LIVA SMART HEALTHCARE ” has been submitted for the Bachelor of Computer Science and Engineering, Poornima Institute Of Engineering & Technology, Jaipur during the academic year 2018-2019 is a bonafide piece of project work carried out by “ **Ishan Verma, Akshita Rajoria & Laveena Chaturvedi**” towards the partial fulfillment for the award of the Degree (B.Tech.) under the guidance of “**Mr. Ajay Saini**” and supervision and no part of thereof has been submitted by them for any degree or diploma.

Project Guide

Project Coordinator

Mr. Deepak Moud

Mr. Ajay Saini

Prof. (Dr.) Praveen Gupta

(H.O.D CSE)

(Assistant Professor)

(Professor)

CANDIDATE'S DECLARATION

We, **Ishan Verma (PIET15CE044), Akshita Rajoria (PIET15CE013) & Laveena Chaturvedi (PIET15CE054)** B.Tech (Semester- VIII) of “**Poornima Institute Of Engineering & Technology, Jaipur**” hereby declare that the Project Report entitled “ **LIVA SMART HEALTHCARE** ” is an original work and data provided in the study is authentic to the best of our knowledge. This report has not been submitted to any other Institute for the award of any other degree.

ISHAN VERMA
(PIET15CE044)

AKSHITA RAJORIA
(PIET15CE013)

LAVEENA CHATURVEDI
(PIET15CE054)

Place: JAIPUR

Date: 18th March 2019

ACKNOWLEDGMENT

It is our pleasure to be indebted to various people, who directly or indirectly contributed in the development of this work and who influenced our thinking, behavior and acts during the course of study.

We express our sincere gratitude to ***Dr. Dinesh Goyal***, Director, PIET for providing us an opportunity to undergo this Major Project as the part of the curriculum.

We are thankful to ***Mr. Deepak Moud, HOD, CS*** for his support, cooperation, and motivation provided to us during the training for constant inspiration, presence and blessings.

We are thankful to ***Mr. Ajay Saini*** for his support, cooperation, and motivation provided to us during the training for constant inspiration, presence and blessings.

We also extend our sincere appreciation to ***Prof. (Dr.) Praveen Gupta*** who provided his valuable suggestions and precious time in accomplishing our Project report.

Lastly, we would like to thank the almighty and our parents for their moral support and friends with whom we shared our day-to-day experience and received lots of suggestions that improved our quality of work.

Ishan Verma

(PIET15CE044)

Akshita Rajoria

(PIET15CE013)

Laveena Chaturvedi

(PIET15CE054)

TABLE OF CONTENTS

LIST OF FIGURES	2
LIST OF TABLES	2
ABSTRACT	4
KEYWORDS:.....	4
CHAPTER 1 (Introduction).....	5
CHAPTER 2 (Product Backlog).....	16
CHAPTER 3 (Technology applied & Project Mnagement).....	24
CHAPTER 4 (Project Implementation)	41
CHAPTER 5 (Conclusion)	46
ANNEXURES	48

List of Figures

Figure 1 Smart Healthcare System	5
Figure 2 Basic Architecture	7
Figure 3 NodeMCU (Labelled).....	12
Figure 4 NodeMCU (Front).....	13
Figure 5 NodeMCU (Back)	13
Figure 6 NodeMCU.....	13
Figure 7 NodeMCU GPIO Pins	14
Figure 8 Micropython Code I.....	41
Figure 9 Micropython Code II.....	41
Figure 10 Home Page.....	41
Figure 11 Register Page	42
Figure 12 Patient Registration Page.....	42
Figure 13 Doctor Registration Page.....	43
Figure 14 Patient List	43
Figure 15 Patient Data.....	44
Figure 16 MySQL Database	44

List of Tables

Table 1 Product Backlog	18
Table 2 Sprint Backlog 1	20
Table 3 Sprint Backlog 2.....	21
Table 4 Sprint Backlog 3.....	22
Table 5 Sprint Backlog 4.....	23

ABSTRACT

This report is published to show the importance of IoT in healthcare nowadays. Technology plays an important role in this field not only for devices based on sensors but also in communicating, recording and various other fields too. So, we used this latest trend of using IoT in healthcare with some devices and sensor and proposed a system. The system comprises of a wearable device with a website, the device will send the data from the patient or user with the help of NodeMCU and sensors integrated to it, and the website is used as an interface between the user and doctor where the doctor can observe the patient data. This prior information of patient can be useful to predict the patient health or any variation that will be observed in patient health in the future. The main aim of the system was to reduce the time and headache of the patient to visit the doctor again and again by connecting to the doctors who are far from patients, and immediate action will increase the health rate among people. The proposed system will provide better medical services to users by connecting and accumulating data through a device which will include user's heart rate, temperature, blood pressure and sends a notification in form of mail or message to the concerned doctor in emergency cases.

KEYWORDS: IoT, Healthcare, Sensor, NodeMCU

CHAPTER 1

1. INTRODUCTION TO PROJECT

The Internet of things (IoT) is the network of physical devices, vehicles, home appliances, and other items embedded with electronics, software, sensors, actuators, and connectivity which enables these things to connect, collect and exchange data.

IoT involves extending Internet connectivity beyond standard devices, such as desktops, laptops, smartphones, and tablets, to any range of traditionally dumb non-internet-enabled physical devices and everyday objects. Embedded with technology, these devices can communicate and interact over the Internet, and they can be remotely monitored and controlled. With the arrival of driverless vehicles, a branch of IoT, i.e. the Internet of Vehicle starts to gain more attention.

LIVA SMART HEALTH model enables users to improve health-related risks and reduce healthcare costs by collecting, recording, analyzing and sharing large data streams in real time and efficiently. The idea of this project came so to reduce the headache of the patient to visit doctor every time he needs to check his blood pressure, heartbeat rate, temperature etc. With the help of this proposal, the time of both patients and doctors are saved and doctors can also help in the emergency scenario as much as possible.

The proposed technique has many advantages like:

- Reduce the headache of visiting the doctor again and again
- A handy device through which health can be analyzed.
- Efficient design
- Fast response
- User friendly

1.1 Aim

LIVA SMART HEALTH model enables users to improve health-related risks and reduce healthcare costs by collecting, recording, analyzing and sharing large data streams in real time and efficiently.

1.2 Objectives

- Reduce the headache of visiting the doctor again and again
- collecting, recording, analyzing and sharing large data streams in real time and efficiently

1.3 Problem Statement

LIVA Smart health lets you automatically log your medical data through Bluetooth/Wi-Fi entitled devices. It takes note of your health by storing, analyzing and sharing your medical records. It also advises you on the smart tips and services based upon your health analysis. It also gives you alerts and messages about your health risks. It enables you to remotely monitor the health reports as well as also has the option of connecting yourself to various health service providers like pharma, labs, home care, and teleconsulting.

1.3.1 Block Diagram

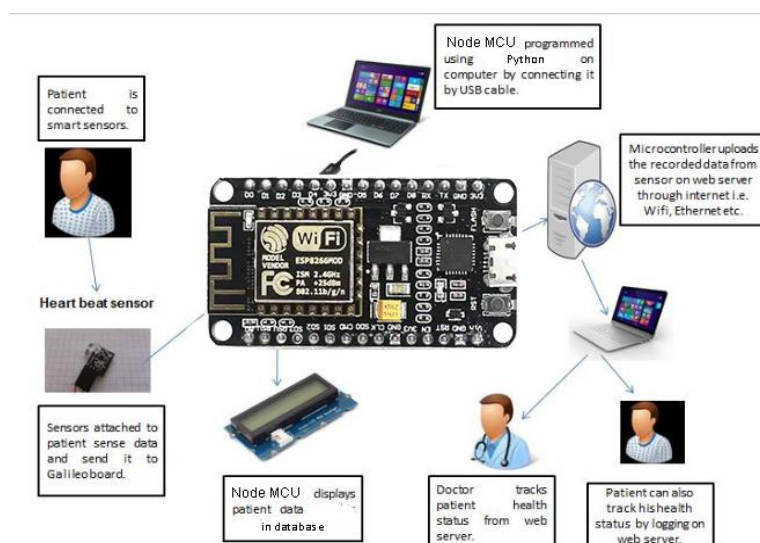


Figure.1 SMART HEALTHCARE SYSTEM (LIVA SMART HEALTH)

1.3.2 Proposed Model

The proposed model enables users to improve health-related risks and reduce healthcare costs by collecting, recording, analyzing and sharing large data streams in real time and efficiently. The idea of this project came so to reduce the headache of the patient to visit doctor every time he needs to check his blood pressure, heartbeat rate, temperature etc. With the help of this proposal, the time of both patients and doctors are saved and doctors can also help in the emergency scenario as much as possible.

1.3.3 Proposed Outcome

The proposed outcome of the project is to give proper and efficient medical services to patients by connecting and collecting data information through health status monitors which would include patient's heart rate, blood pressure, and ECG and sends an emergency alert to patient's doctor with his current status and full medical information.

LIVA Smart health lets you automatically log your medical data through Bluetooth/Wi-Fi entitled devices. It takes note of your health by storing, analyzing and sharing your medical records. It also advises you on the smart tips and services based upon your health analysis. It also gives you alerts and messages about your health risks. It enables you to remotely monitor the health reports as well as also has the option of connecting yourself to various health service providers like pharma, labs, home care, and teleconsulting.

We have proposed a robust health monitoring system that is intelligent enough to monitor the patient automatically using IOT that collects the status information through these systems which would include patient's heart rate, blood pressure, and ECG and sends an emergency alert to patient's doctor with his current status and full medical information. This would help the doctor to monitor his patient from anywhere and also to the patient to send his health status directly without visiting the hospital.

Our model can be deployed at various hospitals and medical institutes. The system uses smart sensors that generate raw data information collected from each sensor and send it to a database server where the data can be further analyzed and statistically maintained to be used by the medical experts. Maintaining a database server is a must so that there is even track of previous medical record of the patient providing a better and improved examining.

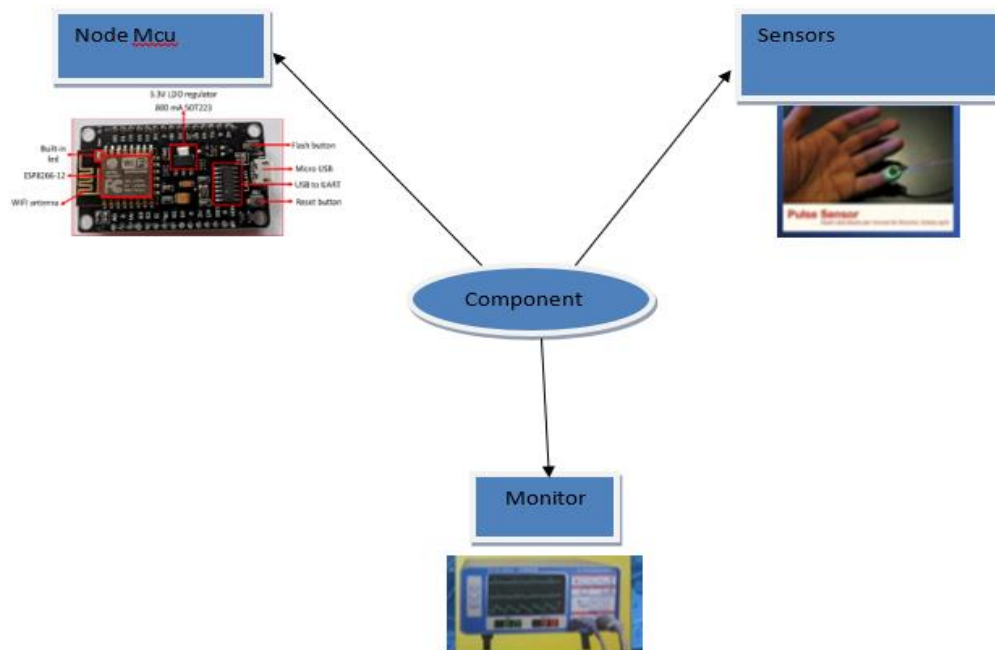


Figure.2 Basic architecture

The brain of our model is the NodeMCU, Using MicroPython is a great way to get the most of your ESP8266 board. And vice versa, the ESP8266 chip is a great platform for using MicroPython. This tutorial will guide you through setting up MicroPython, getting a prompt, using WebREPL, connecting to the network and communicating with the Internet, using the hardware peripherals, and controlling some external components.

The model is finally deployed over a normal fit person and her heart rate and temperature details are plotted on a real-time graph. An example output of a proposed health monitor device is shown in which the patients' personal details are shown and alongside her live heart rate and body temperature is being traced in real time.

1.4 Software Requirements

The first thing you need is a board with an ESP8266 chip. The MicroPython software supports the ESP8266 chip itself and any board should work. The main characteristic of a board is how much flash it has, how the GPIO pins are connected to the outside world, and whether it includes a built-in USB-serial converter to make the UART available to your PC.

The minimum requirement for flash size is 1Mbyte. There is also a special build for boards with 512KB, but it is highly limited comparing to the normal build: there is no support for the filesystem, and thus features which depend on it won't work (WebREPL, up, etc.). As such, the 512KB build will be more interesting for users who build from source and fine-tune parameters for their particular application.

Names of pins will be given in this tutorial using the chip names (eg GPIO0) and it should be straightforward to find which pin this corresponds to on your particular board.

Powering the board

If your board has a USB connector on it then most likely it is powered through this when connected to your PC. Otherwise, you will need to power it directly. Please refer to the documentation for your board for further details.

Getting the firmware

The first thing you need to do is download the most recent MicroPython firmware .bin file to load onto your ESP8266 device. You can download it from the MicroPython downloads page. From here, you have 3 main choices

- Stable firmware builds for 1024kb modules and above.
- Daily firmware builds for 1024kb modules and above.
- Daily firmware builds for 512kb modules.

If you are just starting with MicroPython, the best bet is to go for the Stable firmware builds. If you are an advanced, experienced MicroPython ESP8266 user who would like to follow development closely and help with testing new features, there are daily builds

Support for 512kb modules is provided on a feature preview basis. For end users, it's recommended to use modules with a flash of 1024kb or more. As such, only daily builds for 512kb modules are provided.

Deploying the firmware

Once you have the MicroPython firmware (compiled code), you need to load it onto your ESP8266 device. There are two main steps to do this: first you need to put your device in boot-loader mode, and second, you need to copy across the firmware. The exact procedure for these steps is highly dependent on the particular board and you will need to refer to its documentation for details.

If you have a board that has a USB connector, a USB-serial converter, and has the DTR and RTS pins wired in a special way then deploying the firmware should be easy as all steps can be done automatically. Boards that have such features include the Adafruit Feather HUZZAH and NodeMCU boards. For best results, it is recommended to first erase the entire flash of your device before putting on new MicroPython firmware. Currently, we only support esptool.py to copy across the firmware.

```
pip install esptool
```

Versions starting with 1.3 support both Python 2.7 and Python 3.4 (or newer). An older version (at least 1.2.1 is needed) works fine but will require Python 2.7.

Any other flashing program should work, so feel free to try them out or refer to the documentation for your board to see its recommendations.

Using esptool.py you can erase the flash with the command:

```
esptool.py --port /dev/ttyUSB0 erase_flash
```

And then deploy the new firmware using:

```
esptool.py --port /dev/ttyUSB0 --baud 460800 write_flash --flash_size=detect 0 esp8266-20170108-v1.8.7.bin
```

You might need to change the “port” setting to something else relevant for your PC. You may also need to reduce the baud rate if you get errors when flashing (eg down to 115200). The filename of the firmware should also match the file that you have.

For some boards with a particular FlashROM configuration (e.g. some variants of a NodeMCU board) you may need to use the following command to deploy the firmware (note the -fm dio option):

```
esptool.py --port /dev/ttyUSB0 --baud 460800 write_flash --flash_size=detect -fm dio 0  
esp8266-20170108-v1.8.7.bin
```

If the above commands run without error then MicroPython should be installed on your board!

Serial prompt

Once you have the firmware on the device you can access the REPL (Python prompt) over UART0 (GPIO1=TX, GPIO3=RX), which might be connected to a USB-serial converter, depending on your board. The baud rate is 115200. The next part of the tutorial will discuss the prompt in more detail.

WiFi

After a fresh install and boot the device configures itself as a Wi-Fi access point (AP) that you can connect to. The ESSID is of the form MicroPython-xxxxxx where the x's are replaced with part of the MAC address of your device (so will be the same every time, and most likely different for all ESP8266 chips). The password for the WIFI is micropythoN (note the upper-case N). Its IP address will be 192.168.4.1 once you connect to its network. Wi-Fi configuration will be discussed in more detail later in the tutorial.

Troubleshooting installation problems

If you experience problems during flashing or with running firmware immediately after it, here are troubleshooting recommendations:

- Be aware of and try to exclude hardware problems. There are 2 common problems: bad power source quality and worn-out/defective FlashROM. Speaking of the power source, not just raw amperage is important, but also low ripple and noise/EMI in general. If you experience issues with self-made or wall-wart style power supply, try USB power from a computer. Unearthed power supplies are also known to cause problems as they source of increased EMI (electromagnetic interference) - at the very least, and may lead to electrical devices breakdown. So, you are advised to avoid using unearthed power connections when working with ESP8266 and other boards. In regard to FlashROM hardware problems, there are independent (not related to MicroPython in any way) reports (e.g.) that on some ESP8266 modules, FlashROM can be programmed as little as 20 times before programming errors occur. This is *much* less than 100,000 programming cycles cited for

FlashROM chips of a type used with ESP8266 by reputable vendors, which points to either production rejects, or second-hand worn-out flash chips to be used on some (apparently cheap) modules/boards. You may want to use your best judgment about the source, price, documentation, warranty, post-sales support for the modules/boards you purchase.

- The flashing instructions above using flashing speed of 460800 baud, which is a good compromise between speed and stability. However, depending on your module/board, USB-UART converter, cables, host OS, etc., the above baud rate may be too high and lead to errors. Try a more common 115200 baud rate instead of in such cases.
- If a lower baud rate didn't help, you may want to try an older version of esptool.py, which had a different programming algorithm:
- `pip install esptool==1.0.1`
This version doesn't support `--flash_size=detect` option, so you will need to specify FlashROM size explicitly (in megabits). It also requires Python 2.7, so you may need to use `pip2` instead of `pip` in the command above.
- The `--flash_size` option in the commands above is mandatory. Omitting it will lead to corrupted firmware.
- To catch incorrect flash content (e.g. from a defective sector on a chip), add `--verify` switch to the commands above.
- Additionally, you can check the firmware integrity from a MicroPython REPL prompt (assuming you were able to flash it and `--verify` option doesn't report errors):
- `import esp`
- `esp.check_fw()`
If the last output value is `True`, the firmware is OK. Otherwise, it's corrupted and needs to be reflashed correctly.
- If you experience any issues with another flashing application (not `esptool.py`), try `esptool.py`, it is a generally accepted flashing application in the ESP8266 community.
- If you still experience problems with even flashing the firmware, please refer to `esptool.py` project page, <https://github.com/espressif/esptool> for additional documentation and bug tracker where you can report problems.
- If you are able to flash the firmware, but `--verify` option or `esp.check_fw()` return errors even after multiple retries, you may have a defective FlashROM chip, as explained above.

1.5 Hardware Requirements

The NodeMcu IoT Computer:

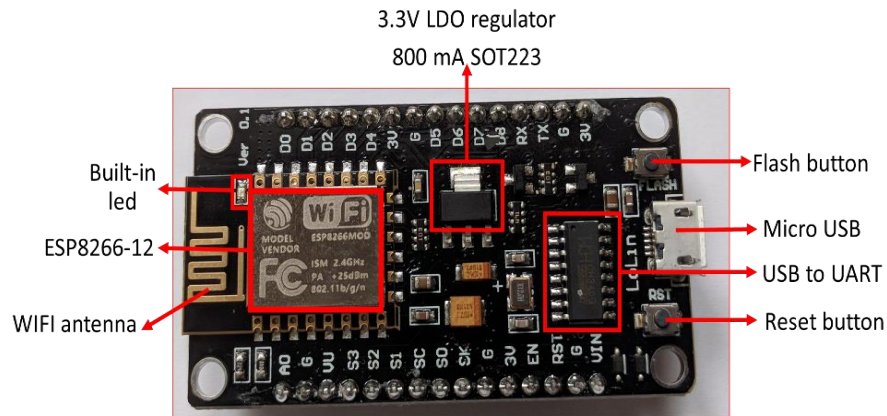


Figure.3 Node MCU (Labelled)

The Specs:

	NodeMcu
Processor	L106 32-bit RISC
Memory	128Kbytes
Storage	4Mbytes
Input	16 GPIO pins
Type	32-bit microcontroller
WiFi adapter	b/g/n Wi-Fi
Operating System	XTOS

NodeMCU devkit is a development kit for NodeMCU firmware. Its aim is to make NodeMCU firmware development easier. With a micro USB cable, you can connect NodeMCU devkit to your laptop and flash it without any trouble, just like Arduino. It is open hardware, with ESP-12 core with 32Mbits(4MBytes) flash.

Devkit 0.8

This was the original design of NodeMCU devkit which never released on general sale.

Devkit 0.9

It is the second design of NodeMCU devkit. It uses CH340G as UART bridge and can flash firmware automatically by using nodemcu-flasher.



Figure.4 Node MCU (Front)



Figure.5 NodeMCU (Back)

Devkit 1.0

It is the 5th design of NodeMCU devkit. This uses CP2102 as UART bridge and can flash firmware automatically by using nodemcu-flasher. It also supports Apple's MAC OS.

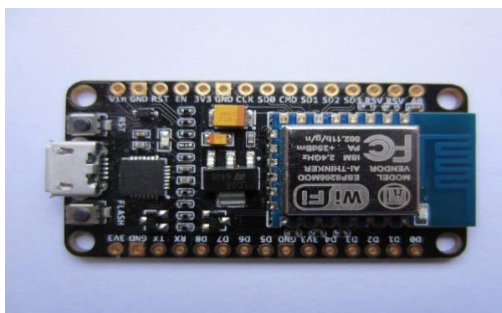


Figure.6 Node MCU

GPIOs

The NodeMcu has sixteen General Purpose Input/Output (GPIO) pins that can be controlled by the user.

GPIO Mapping

On the NodeMcu, the GPIOs are laid out in two banks:

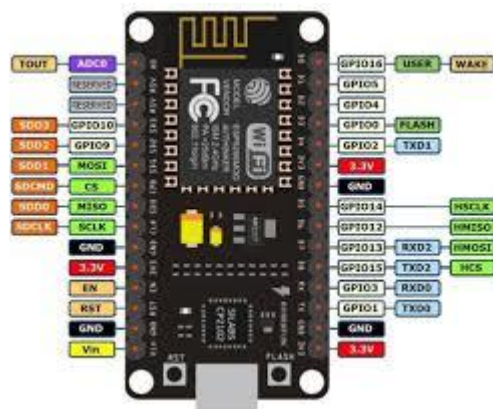


Figure.7 NodeMCU GPIO Pins

PRODUCT BACKLOG

1. Product Backlog

The product backlog comprises an ordered list of product requirements that a scrum team maintains for a product. The format of product backlog items varies, common formats include user stories, use cases, or any other requirements format the team finds useful. These will define features, bug fixes, non-functional requirements, etc.—whatever must be done to successfully deliver a viable product. The product owner prioritizes product backlog items (PBIs) based on considerations such as risk, business value, dependencies, size, and date needed.

The product backlog is what will be delivered, ordered into the sequence in which it should be delivered. It is visible to everyone but may only be changed with the consent of the product owner, who is ultimately responsible for ordering product backlog items for the development team to choose.

The product backlog contains the product owner's assessment of business value and the development team's assessment of development effort, which are often, but not always, stated in story points using the rounded Fibonacci scale. These estimates help the product owner to gauge the timeline and may influence ordering of product backlog items; for example, if two features have the same business value, the product owner may schedule earlier delivery of the one with the lower development effort (because the return on investment is higher) or the one with higher development effort (because it is more complex or riskier, and they want to retire that risk earlier).

The product backlog and the business value of each product backlog item is the responsibility of the product owner. The size (i.e. estimated complexity or effort) of each item is, however, determined by the development team, who contributes by sizing in story points or in estimated hours.

Scrum advocates that the role of product owner is assigned. The product owner is responsible for maximizing the value of the product. The product owner gathers input and takes feedback from, and is lobbied by, many people, but ultimately makes the call on what gets built.

The product backlog:

- Captures requests to modify a product—including new features, replacing old features, removing features, and fixing issues
- Ensures the development team has work that maximizes business benefit to the product owner

Typically, the product owner and the scrum team come together and write down everything that must be prioritized, and this becomes content for the first sprint—which is a block of time meant for focused work on selected items that can be accommodated within a timeframe. The product backlog can evolve as new information surfaces about the product and about its customers, and so later sprints may address new work.

The following items typically comprise a product backlog: features, bugs, technical work, and knowledge acquisition. A feature is wanted, while a bug is unintended or unwanted (but may not be necessarily something defective). An example of technical work could be to run a virus check on all developers' workstations. An example of knowledge acquisition could be to research Wordpress plugin libraries and making a selection.

SPRINT BACKLOG	US ID	BACKLOG ITEM			PRIORIT Y
		AS A/AN	I WANT TO	SO THAT	
1	SB1/US 1	Patient	Register Myself Using Mail Id	I Can Receive A Confirmation/Verification Mail	1
1	SB1/US 2	Patient	Receive A Confirmation Mail	I Can Access The Health Analysis Web Page	1
1	SB1/US 3	Patient	Login Into The System Using Mail Id And Password	I Can View/Edit My Profile	1
1	SB1/US 4	Patient	Modify The Account Password	I Can Keep My Account Secure	1
1	SB1/US 5	Doctor	Register Myself Using Mail Id	I Can Receive A Confirmation/Verification Mail	1
1	SB1/US 6	Doctor	Receive A Confirmation Mail	I Can Access The Health Analysis Web Page	1
1	SB1/US 7	Doctor	Login Into The System Using Mail Id And Password	I Can View/Edit My Profile	1

1	SB1/US 8	Doctor	Modify The Account Password	I Can Keep My Account Secure	1
1	SB1/US 9	Hospitals	Login Into The System Using Mail Id And Password	I Can View/Edit My Profile	1
1	SB1/US 10	Hospitals	Modify The Account Password	I Can Keep My Account Secure	1
1	SB1/US 11	Hospitals	Manage The Patient Activities	I Can Manage Patient Activities	1
2	SB2/US 12	Admin	purchase a NodeMcu Device	I Can do coding in it	2
2	SB2/US 13	Admin	assemble the NodeMcu for Sensor(Blood Pressure)	I Can Analyze the Performance	2
2	SB2/US 14	Admin	assemble the NodeMcu for Sensor(Heart beat)	I Can Analyze the Performance	2
2	SB2/US 15	Admin	assemble the NodeMcu for Sensor(Temperature)	I Can Analyze the Performance	2
2	SB2/US 16	Admin	assemble the NodeMcu for Sensor glucose level)	I Can Analyze the Performance	2
2	SB2/US 17	Admin	Assemble the device	I Can make them in a very comfort and perfect way	2
2	SB2/US 18	Admin	do programming	I Can program to transfer data to a website	2
3	SB3/US 19	Patient	View The Report	I Can take precautions	3
3	SB3/US 20	Patient	View the daily report	I Can work accordingly	3
3	SB3/US 21	Patient	View My device problem	I Can Prevent The Mistakes	3
3	SB3/US 22	Patient	Download report	I Can show it wherever I want	3
3	SB3/US 23	Doctor	View The Report of patient	I Can analyze the patient	3
3	SB3/US 24	Doctor	View the daily report	I Can work accordingly and give prescriptions	3
3	SB3/US 25	Admin	Testing	I Can check my Device	3
3	SB3/US 26	Admin	Testing	I Can check the errors	3
3	SB3/US 27	Web Admin	receive data from NodeMcu	I Can view the reports	3
3	SB3/US 28	Web Admin	Download patient's reports	I Can Complete And Submit Them On Time	3
3	SB3/US 29	Web Admin	Fire Insert Queries	I Can Insert New Records In The Database	3
3	SB3/US 30	Web Admin	Fire Select Queries	I Can View The Records	3
3	SB3/US 31	Web Admin	Fire Update Queries	I Can Keep The Database Updated	3
3	SB3/US 32	Web Admin	Fire Delete Queries	I Can Remove Unwanted Data	3
3	SB3/US 33	Web Admin	Take Backup	I Can Save The Old Records In The Database For Future Use	3
3	SB3/US 34	Web Admin	Delete Backup	I Can Delete The Records Which Is Not Needed Any More	3

3	SB3/US 35	Web Admin	Generate Report	I Can review patient	3
3	SB3/US 36	Web Admin	Interpret data	I Can view data	3
3	SB3/US 37	Web Admin	Analysis of data	I Can analysis data	3
3	SB3/US 38	Web Admin	Represent data	I Can represent data on a graph	3
3	SB3/US 39	Hospita ls	Create A List Of Patients using the device	I Can Create A List Of Patients using the device	3
3	SB3/US 40	Hospita ls	Sending Information Through Emails	I Can Send Information Through Emails	3
3	SB3/US 41	Hospita ls	Manage The Time Schedule For Consultation	I Can Manage The Time Schedule For Consultation	3
3	SB3/US 42	Hospita ls	View The Details Of All Patients	I Can View All The Details Of Patients	3
3	SB3/US 43	Hospita ls	Prepare Prescription For All Patients	I Can Prepare Prescription For All Patients	3
3	SB3/US 44	Hospita ls	Update Time For Consultation	I Can Update Time For Consultation	3
3	SB3/US 45	Hospita ls	Access To All Information Of Doctors On Duties	I Can Access The Information Of Doctors On Duties	3
3	SB3/US 46	Hospita ls	Manage All Statistics And Data About Patients	I Can Manage All Statistics And Data About Patients	3
3	SB3/US 47	Hospita ls	Generate receipt for consultation	I Can assist doctor to prescribe medicines on the receipt	3
4	SB4/US 48	Patient	Discuss Among Other People	I Can Share My Experience	4
4	SB4/US 49	Doctor	Generate bill for Medicines prescribed	I Can take medicine charges	4
4	SB4/US 50	Doctor	Discuss Among Other People	I Can Share My Experience	4
4	SB4/US 51	Hospita ls	Review And Feedback From Patients By Email	I Can Review And Feedback From Patients By Email	4
4	SB4/US 52	Hospita ls	Generate bill for doctor's fees	I Can take consultation charges	4
4	SB4/US 53	Hospita ls	Generate bill for Medicines prescribed	I Can take medicine charges	4

Table 1 Product Backlog

2. Sprint Backlog

The sprint backlog is the list of work the development team must address during the next sprint.^[33] The list is derived by the scrum team progressively selecting product backlog items in priority order from the top of the product backlog until they feel they

have enough work to fill the sprint. The development team should keep in mind its past performance assessing its capacity for the new sprint, and use this as a guide line of how much 'effort' they can complete.

The product backlog items may be broken down into tasks by the development team.^[33] Tasks on the sprint backlog are never assigned; rather, tasks are signed up for by the team members as needed according to the set priority and the skills of the team. This promotes self-organization of the development team and developer buy-in.

The sprint backlog is the property of the development team, and all included estimates are provided by the development team. Often an accompanying task board is used to see and change the state of the tasks of the current sprint, like to do, in progress and done.

Once a sprint backlog is committed, no additional work can be added to the sprint backlog except by the team. Once a sprint has been delivered, the product backlog is analyzed and reprioritized if necessary, and the next set of functionality is selected for the next sprint.

2.1 Sprint Backlog-1

US ID	USER STORY	TASK ID	TASKS
SB1/US1	Register Myself Using Mail Id	SB1/D1/T1	Designing of GUI
		SB1/D1/T2	Validating user input at the client level
		SB1/D1/T3	Validating user input at server level
		SB1/D1/T4	Connection with Database to insert the record.
		SB1/D1/T5	Testing of the module on localhost and then on the server.
SB1/US2	Receive A Confirmation Mail	SB1/D2/T1	Designing of Email(to be send)
		SB1/D2/T2	Send Email via coding using the Mandrill App
		SB1/D2/T3	Connection with the database to update the status in the database.
		SB1/D2/T4	Testing of the module on localhost and then on the server.
SB1/US3	Login Into The System Using Mail Id And Password	SB1/D3/T1	Designing of GUI
		SB1/D3/T2	Validating user input at the client level
		SB1/D3/T3	Validating user input at server level
		SB1/D3/T4	Connection with Database to insert the record.
		SB1/D3/T5	Testing of the module on localhost and then on the server.

SB1/US4	Modify The Account Password	SB1/D4/T1	Designing of GUI
		SB1/D4/T2	Validating user input at the client level
		SB1/D4/T3	Validating user input at server level
		SB1/D4/T4	Connection with Database to insert the record.
		SB1/D4/T5	Testing of the module on localhost and then on the server.
SB1/US5	Register Myself Using Mail Id	SB1/D5/T1	Designing of GUI
		SB1/D5/T2	Validating user input at the client level
		SB1/D5/T3	Validating user input at server level
		SB1/D5/T4	Connection with Database to insert the record.
		SB1/D5/T5	Testing of the module on localhost and then on the server.
SB1/US6	Receive A Confirmation Mail	SB1/D6/T1	Designing of Email(to be send)
		SB1/D6/T2	Send Email via coding using the Mandrill App
		SB1/D6/T3	Connection with the database to update the status in the database.
		SB1/D6/T4	Testing of the module on localhost and then on the server.
SB1/US7	Login Into The System Using Mail Id And Password	SB1/D7/T1	Designing of GUI
		SB1/D7/T2	Validating user input at the client level
		SB1/D7/T3	Validating user input at server level
		SB1/D7/T4	Connection with Database to insert the record.
		SB1/D7/T5	Testing of the module on localhost and then on the server.
SB1/US8	Modify The Account Password	SB1/D8/T1	Designing of GUI
		SB1/D8/T2	Validating user input at the client level
		SB1/D8/T3	Validating user input at server level
		SB1/D8/T4	Connection with Database to insert the record.
		SB1/D8/T5	Testing of the module on localhost and then on the server.
SB1/US9	Login Into The System Using Mail Id And Password	SB1/D9/T1	Designing of GUI
		SB1/D9/T2	Validating user input at the client level
		SB1/D9/T3	Validating user input at server level
		SB1/D9/T4	Connection with Database to insert the record.
		SB1/D9/T5	Testing of the module on localhost and then on the server.
SB1/US10	Modify The Account Password	SB1/D10/T1	Designing of GUI
		SB1/D10/T2	Validating user input at the client level
		SB1/D10/T3	Validating user input at server level
		SB1/D10/T4	Connection with Database to insert the record.
		SB1/D10/T5	Testing of the module on localhost and then on the server.
SB1/US11	Manage The Patient Activities	SB1/D11/T1	Designing of GUI
		SB1/D11/T2	Validating user input at the client level
		SB1/D11/T3	Validating user input at server level
		SB1/D11/T4	Connection with Database to insert the record.

		SB1/D11/T5	Testing of the module on localhost and then on the server.
--	--	------------	--

Table 2 Sprint Backlog 1

2.2 Sprint Backlog-2

US ID	USER STORY	TASK ID	TASKS
SB2/US12	As an Admin I want to purchase a NodeMcu Device	SB2/D1/T1	Search about the hardware setup
		SB2/D1/T2	Search about components required
		SB2/D1/T3	Purchase component from Market
		SB2/D1/T4	Develop the hardware
SB2/US13	As an Admin I want to assemble the NodeMcu for Sensor(Blood Pressure)	SB2/D2/T1	Search about the hardware setup
		SB2/D2/T2	Purchase component from Market
		SB2/D2/T3	Develop the hardware
SB2/US14	As an Admin I want to assemble the NodeMcu for Sensor(Heart beat)	SB2/D3/T1	Search about the hardware setup
		SB2/D3/T2	Purchase component from Market
		SB2/D3/T3	Develop the hardware
SB2/US15	As an Admin I want to assemble the NodeMcu for Sensor(Temperature)	SB2/D4/T1	Search about the hardware setup
		SB2/D4/T2	Purchase component from Market
		SB2/D4/T3	Develop the hardware
SB2/US16	As an Admin I want to assemble the NodeMcu for Sensor(glucose level)	SB2/D5/T1	Search about the hardware setup
		SB2/D5/T2	Purchase component from Market
		SB2/D5/T3	Develop the hardware
SB2/US17	As an Admin I want to Assemble the device	SB2/D6/T1	Search about the hardware setup
		SB2/D6/T2	Assemble all the sensors and NodeMcu
		SB2/D6/T3	Develop the hardware
SB2/US18	As an Admin I want to do programming	SB2/D7/T1	coding of onscreen application
		SB2/D7/T2	coding of onscreen application
		SB2/D7/T3	Debug the code
		SB2/D7/T4	Debug the code

Table 3: Sprint Backlog 2

2.3 Sprint Backlog-3

US ID	USER STORY	TASK ID	TASKS
SB3/US19	As a patient, I want to View The Report	SB3/D1/T1	login with the registered email id
		SB3/D1/T2	view and analyze the report
SB3/US20	As a patient, I want to View the daily report	SB3/D2/T1	login with the registered email id
		SB3/D2/T2	view and analyze the report
SB3/US21	As a patient, I want to View My device problem	SB3/D3/T1	use the device
		SB3/D3/T2	observe the results
		SB3/D3/T3	if problem found the report to admin
SB3/US22	As a patient, I want to Download the report	SB3/D4/T1	login with the registered email id
		SB3/D4/T2	download the report
SB3/US23	As a doctor, I want to View The Report of patient	SB3/D5/T1	login with registered email id
		SB3/D5/T2	select the patient associated
		SB3/D5/T3	view the report of a patient
SB3/US24	As a doctor, I want to View the daily report	SB3/D6/T1	login with registered email id
		SB3/D6/T2	select the patient associated
		SB3/D6/T3	view the report of a patient
SB3/US25	As an admin, I want to do Testing of the device	SB3/D7/T1	view the complaints posted by patients
		SB3/D7/T2	test the device
		SB3/D7/T3	modify the device
SB3/US26	As an admin, I want to do Testing of program	SB3/D8/T1	view the complaints posted by patients
		SB3/D8/T2	test the program
		SB3/D8/T3	modify the program

Table 4: Sprint Backlog 3

2.4 Sprint Backlog-4

US ID	USER STORY	TASK ID	TASKS
SB4/US48	Discuss Among Other People	SB4/D1/T1	share the device information
		SB4/D1/T2	share the application information
SB4/US49	Generate bill for Medicines prescribed	SB4/D2/T1	analyze the patient problem
		SB4/D2/T2	prescribe the medicine
		SB4/D2/T3	generate a bill for the prescription
SB4/US50	Discuss Among Other People	SB4/D3/T1	share the device information
		SB4/D3/T2	share the application information
SB4/US51		SB4/D4/T1	use the device

	Review And Feedback From Patients By Email	SB4/D4/T2	login by using registered email id
		SB4/D4/T3	provide review and feedback
SB4/US52	Generate bill for doctor's fees	SB4/D5/T1	analyze the patient problem
		SB4/D5/T2	prescribe the medicine
		SB4/D5/T3	generate a bill for the prescription
SB4/US53	Generate bill for Medicines prescribed	SB4/D5/T1	analyze the patient problem
		SB4/D5/T2	prescribe the medicine
		SB4/D5/T3	generate a bill for the prescription

Table 5: Sprint Backlog 4

CHAPTER 3

TECHNOLOGY APPLIED AND PROJECT MANAGEMENT

3.1 Description of Technologies

- **Data acquisition technologies:** this category contains all surveying, mapping, navigation and sensing technologies.
- **Data analysis and eva Micropythontion technologies:** these technologies range from simple computer-based decision models to complex health management & information systems including many different variables.
- **Precision application technologies:** this category contains all application technologies, focusing on variable-rate application and guidance technologies. Each technology referred to in the table above will be analyzed in this chapter.

3.1.1 Other technologies: -

- **Php**
- **Python**
- **IOT (Internet of things)**

Php: - It is a technology which is used to designed and develop website for connecting sensors and software to the server.

Python: - is an interpreted high-level programming language for general-purpose programming. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

Iot: - It is technology that user Internet as a server to send and receive data from the sensor.

3.2 Project management :

Project management is the application of processes, methods, knowledge, skills, and experience to achieve the project objectives. General. A project is a unique, transient endeavor, undertaken to achieve planned objectives, which could be defined in terms of outputs, outcomes or benefits.

Project management is the practice of initiating, planning, executing, controlling, and closing the work of a team to achieve specific goals and meet specific success criteria at the specified time. A project is a temporary endeavor designed to produce a unique product, service or result with a defined beginning and end undertaken to meet unique goals and objectives, typically to bring about beneficial change or added value. The temporary nature of projects stands in contrast with business as usual, which are repetitive, permanent, or semi-permanent functional activities to produce products or services. In practice, the management of such distinct production approaches requires the development of distinct technical skills and management strategies.

Software project management

Software project management is the art and science of planning and leading software projects. It is a sub-discipline of project management in which software projects are planned, implemented, monitored and controlled.

The job pattern of an IT company engaged in software development can be seen split into two parts:

- Software Creation
- Software Project Management

A project is a well-defined task, which is a collection of several operations done in order to achieve a goal (for example, software development and delivery). A Project can be characterized as:

- Every project may have a unique and distinct goal.
- The project is not a routine activity or day-to-day operations.
- The project comes with a start time and end time.

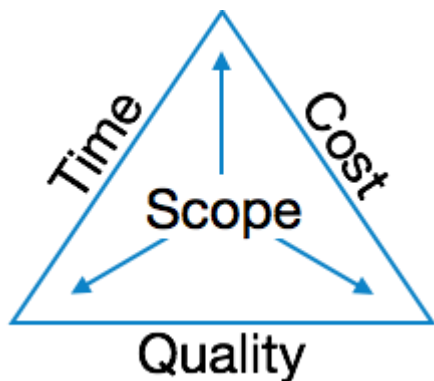
- The project ends when its goal is achieved hence it is a temporary phase in the lifetime of an organization.
- The project needs adequate resources in terms of time, manpower, finance, material and knowledge-bank.

Software Project

A Software Project is the complete procedure of software development from requirement gathering to testing and maintenance, carried out according to the execution methodologies, in a specified period of time to achieve intended software product.

The need for software project management

The software is said to be an intangible product. Software development is a kind of all new stream in world business and there's very little experience in building software products. Most software products are tailor-made to fit the client's requirements. The most important is that the underlying technology changes and advances so frequently and rapidly that the experience of one product may not be applied to the other one. All such business and environmental constraints bring risk in software development hence it is essential to manage software projects efficiently.



The image above shows triple constraints for software projects. It is an essential part of software organization to deliver a quality product, keeping the cost within the client's budget constraints and deliver the project as per schedule. There are several factors, both internal and external, which may impact this triple constrain triangle. Any of three-factor can severely impact the other two.

Therefore, software project management is essential to incorporate user requirements along with budget and time constraints.

Software Project Manager

A software project manager is a person who undertakes the responsibility of executing the software project. The software project manager is thoroughly aware of all the phases of SDLC that the software would go through. The project manager may never directly involve in producing the end product but he controls and manages the activities involved in the production.

A project manager closely monitors the development process, prepares and executes various plans, arranges necessary and adequate resources, maintains communication among all team members in order to address issues of cost, budget, resources, time, quality and customer satisfaction.

Let us see a few responsibilities that a project manager shoulders -

Managing People

- Act as project leader
- Liaison with stakeholders
- Managing human resources
- Setting up reporting hierarchy etc.

Managing Project

- Defining and setting up project scope
- Managing project management activities
- Monitoring progress and performance
- Risk analysis at every phase
- Take the necessary step to avoid or come out of problems
- Act as project spokesperson

Software Management Activities

Software project management comprises of a number of activities, which contains planning of the project, deciding the scope of a software product, estimation of cost in various terms, scheduling of tasks and events, and resource management. Project management activities may include:

- **Project Planning**
- **Scope Management**
- **Project Estimation**

Project Planning

Software project planning is a task, which is performed before the production of software actually starts. It is there for the software production but involves no concrete activity that has any direction connection with software production; rather it is a set of multiple processes, which facilitates software production. Project planning may include the following:

Scope Management

It defines the scope of the project; this includes all the activities, the process needs to be done in order to make a deliverable software product. Scope management is essential because it creates boundaries of the project by clearly defining what would be done in the project and what would not be done. This makes a project to contain limited and quantifiable tasks, which can easily be documented and in turn avoids cost and time overrun.

During Project Scope management, it is necessary to -

- Define the scope
- Decide its verification and control
- Divide the project into various smaller parts for ease of management.
- Verify the scope
- Control the scope by incorporating changes to the scope

Project Estimation

For an effective management accurate estimation of various measures is a must. With correct estimation, managers can manage and control the project more efficiently and effectively.

Project estimation may involve the following:

- **Software size estimation**

Software size may be estimated either in terms of KLOC (Kilo Line of Code) or by calculating a number of function points in the software. Lines of code depend upon coding practices and Function points vary according to the user or software requirement.

- **Effort estimation**

The managers estimate efforts in terms of personnel requirement and man-hour required to produce the software. For effort estimation software size should be known. This can either be derived by managers' experience, the organization's historical data or software size can be converted into efforts by using some standard formulae.

- **Time estimation**

Once size and efforts are estimated, the time required to produce the software can be estimated. An effort required is segregated into sub categories as per the requirement specifications and interdependency of various components of software. Software tasks are divided into smaller tasks, activities or events by Work Breakthrough Structure (WBS). The tasks are scheduled on a day-to-day basis or in calendar months.

The sum of the time required to complete all tasks in hours or days is the total time invested to complete the project.

- **Cost estimation**

This might be considered as the most difficult of all because it depends on more elements than any of the previous ones. For estimating project cost, it is required to consider -

- Size of software
- Software quality
- Hardware
- Additional software or tools, licenses etc.

- Skilled personnel with task-specific skills
- Travel involved
- Communication
- Training and support

Project Estimation Techniques

We discussed various parameters involving project estimation such as size, effort, time and cost. The project manager can estimate the listed factors using two broadly recognized techniques

Decomposition Technique

This technique assumes the software as a product of various compositions.

There are two main models -

- **The line of Code** Estimation is done on behalf of a number of line of codes in the software product.
- **Function Points** Estimation is done on behalf of a number of function points in the software product.

Empirical Estimation Technique

This technique uses empirically derived formulae to make an estimation. These formulae are based on LOC or FPs.

- **Putnam Model**

This model is made by Lawrence H. Putnam, which is based on Norden's frequency distribution (Rayleigh curve). Putnam model maps time and efforts required with software size.

- **COCOMO**

COCOMO stands for constructive cost model, developed by Barry W. Boehm. It divides the software product into three categories of software: organic, semi-detached and embedded.

Project Scheduling

Project Scheduling in a project refers to the roadmap of all activities to be done with specified order and within time slot allotted to each activity. Project managers tend to define various tasks, and project milestones and they arrange them keeping various factors in mind. They look for tasks lie in the critical path in the schedule, which are necessary to complete in a specific manner and strictly within the time allocated. The arrangement of tasks which lies out of critical path is less likely to impact over all schedule of the project.

For scheduling a project, it is necessary to -

- Break down the project tasks into a smaller, manageable form
- Find out various tasks and correlate them
- The estimate time frame required for each task
- Divide time into work-units
- Assign an adequate number of work-units for each task
- Calculate the total time required for the project from start to finish

Resource management

All elements used to develop a software product may be assumed as a resource for that project. This may include human resource, productive tools and software libraries.

The resources are available in limited quantity and stay in the organization as a pool of assets. The shortage of resources hampers the development of the project and it can lag behind the schedule. Allocating extra resources increases development cost in the end. It is, therefore, necessary to estimate and allocate adequate resources for the project.

Resource management includes -

- Defining proper organization project by creating a project team and allocating responsibilities to each team member

- Determining resources required at a particular stage and their availability
- Manage Resources by generating resource request when they are required and de-allocating them when they are no more needed.

Project Risk Management

Risk management involves all activities pertaining to identification, analyzing and making provision for predictable and non-predictable risks in the project. The risk may include the following:

- Experienced staff leaving the project and new staff coming in.
- Change in organizational management.
- Requirement change or misinterpreting requirement.
- Under-estimation of required time and resources.
- Technological changes, environmental changes, business competition.

Risk Management Process

There are following activities involved in the risk management process:

- **Identification** - Make note of all possible risks, which may occur in the project.
- **Categorize** - Categorize known risks into high, medium and low-risk intensity as per their possible impact on the project.
- **Manage** - Analyze the probability of occurrence of risks at various phases. Make a plan to avoid or face risks. Attempt to minimize their side-effects.
- **Monitor** - Closely monitor the potential risks and their early symptoms. Also, monitor the effects of steps taken to mitigate or avoid them.

Project Execution & Monitoring

In this phase, the tasks described in project plans are executed according to their schedules.

Execution needs monitoring in order to check whether everything is going according to the plan. Monitoring is observing to check the probability of risk and taking measures to address the risk or report the status of various tasks.

These measures include -

- **Activity Monitoring** - All activities scheduled within some task can be monitored on a day-to-day basis. When all activities in a task are completed, it is considered complete.
- **Status Reports** - The reports contain the status of activities and tasks completed within a given time frame, generally a week. Status can be marked as finished, pending or work-in-progress etc.
- **Milestones Checklist** - Every project is divided into multiple phases where major tasks are performed (milestones) based on the phases of SDLC. This milestone checklist is prepared once every few weeks and reports the status of milestones.

Project Communication Management

Effective communication plays a vital role in the success of a project. It bridges gaps between the client and the organization, among the team members as well as other stake holders in the project such as hardware suppliers.

Communication can be oral or written. Communication management process may have the following steps:

- **Planning** - This step includes the identifications of all the stakeholders in the project and the mode of communication among them. It also considers if any additional communication facilities are required.
- **Sharing** - After determining various aspects of planning, the manager focuses on sharing correct information with the correct person on the correct time. This keeps everyone involved the project up to date with project progress and its status.
- **Feedback** - Project managers use various measures and feedback mechanism and create status and performance reports. This mechanism ensures that input from various stakeholders is coming to the project manager as their feedback.
- **Closure** - At the end of each major event, end of a phase of SDLC or end of the project itself, administrative closure is formally announced to update every stakeholder by sending email, by distributing a hardcopy of the document or by another mean of effective communication.

After closure, the team moves to the next phase or project.

Configuration Management

Configuration management is a process of tracking and controlling the changes in software in terms of the requirements, design, functions, and development of the product.

IEEE defines it as “the process of identifying and defining the items in the system, controlling the change of these items throughout their life cycle, recording and reporting the status of items and change requests, and verifying the completeness and correctness of items”.

Generally, once the SRS is finalized there is less chance of requirement of changes from the user. If they occur, the changes are addressed only with prior approval of higher management, as there is a possibility of cost and time overrun.

Project management Tools:

Project management required tools to manage the work, time and resources. At present many of the software are available for project management. Some of the popular software tools are as follows.

01. Trello

Trello is a project management tool, instead, this app is a free visual way to glance at the entire project with a single view. With Trello you can organize cards, these cards can be your thoughts, conversations and to-do lists and be placed on a board for everyone to collaborate on.

02. Basecamp

Basecamp is the granddaddy of project management apps. Basecamp is considered the leading project management tool around. It boasts a simple and easy to use interface to collaborate with your team and client. It allows you to create multiple projects and setup discussions, write to-do lists, manage files, create and share documents, and organize dates for scheduling.

03. Teamwork Projects

Teamwork Projects is the ultimate productivity tool to manage projects with your team. Teamwork allows you to keep all your projects, tasks and files all in one place and easily collaborate with a team. Teamwork helps you to visualize the entire project through a marked

calendar and Gantt chart and setup reporting. Teamwork supports file management with Google Drive, Box.com and Dropbox. As well as integration with leading apps such as third-party accounting software and customer support apps.

04. Resource Guru

Billed as the "simple way to schedule people, equipment and other resources", Resource Guru is a streamlined resource scheduling and leave management tool that's designed to keep your projects on track. You can plan your team's workloads, receive daily booking reminders, report on KPIs, and more. Apple, Saatchi & Saatchi and Deloitte are among some of the cloud-based team calendar's heavyweight customers.

05. ActiveCollab

ActiveCollab recently released its new version 5.0. The newly revamped app is now more powerful and focused project management tool. It offers team collaborating features, task management, time tracking and importing expenses. One of the biggest assets of ActiveCollab is it offers invoicing features. You are able to track payments and expenses and have invoices paid directly within ActiveCollab with PayPal, and other credit card payments.

06. Zoho Projects

Zoho offers a wide range of business software including Projects. Zoho Projects is a proficient tool to project plan and project coordinator from start to finish. It boosts all the features you need for project management with some advanced features including reporting, integration with Google Apps and Dropbox, bug tracking, setup Wiki Pages to build a repository of information, forums and more.

07. Jira

Jira is specifically targeted for software development teams. Jira offers abilities to raise issues and bugs. Jira makes it real easy to track bugs and see which issues are still outstanding and how much time was spent on each task. Jira offers other products including Confluence a document collaboration tool, and HipChat a team chat and video and file sharing platform and other products.

08. Asana

Asana is the easiest way for teams to track their work so everyone knows who's doing what, by when. With tasks, projects, conversations, and dashboards, Asana keeps your work organized, and teammates accountable so you can move work forward faster. Asana also lets you keep track of your work wherever you are with mobile apps for both iOS and Android.

09. Podio

Podio is an ever growing tool to organize and communication tool for any business. Podio allows you to personalize this platform to fit your business needs. Besides being able to communicate with a team, setup task management, use as a file storage system, like a traditional project management app, Podio can be an internal intranet for all your colleagues and departments to interact.

10. Freedcamp

Whatever your project may be, either setting up an event, a web project or organizing a wedding, Freedcamp helps you organize and plan effectively. Freedcamp has an organized dashboard to view the entire project at a glance. You can easily setup tasks, use sticky notes to visually setup tasks and organize them into the calendar. Freedcamp provides advance add-ons for high-level business use including CRM, invoicing, issue tracking and setting up wiki pages.

11. Wrike

Wrike is an advance application to help you work smarter. By making sure you are always staying on track and ensure you have the adequate resources to finish on time and on budget. Setting up tasks, engage your team and integrate with your business tools including Google Apps, Microsoft Excel, Dropbox and many more is so easy with Wrike.

PO and Their Relevance to the project

PO1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals and an engineering specialization to the solution of complex engineering problems.

In this project creation process engineering knowledge of software engineering and Electronics, engineering have been applied. we have used software engineering, HTML,XML, java, android, java script, PHP , j2ee, data base, Oracle , my SQL , mango and other

programming language and database to the project. We have applied all the above engineering subjects in our projects.

PO2: Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

In our projects, we have identified a problem, once verified by the client we have worked to identify the solution using all of our theoretical and practical knowledge.

PO3: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis, and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

In the project development, we have applied Integrated Development Environment IDE for the rapid development of the code, used web server for the software development.

PO6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

In 1961, the Conference of Engineering Societies of Western Europe and the United States of America defined "professional engineer" as follows.

A professional engineer is competent by virtue of his/her fundamental education and training to apply the scientific method and outlook to the analysis and solution of engineering problems. He/she is able to assume personal responsibility for the development and application of engineering science and knowledge, notably in research, design, construction, manufacturing, superintending, managing and in the education of the engineer. His/her work is predominantly intellectual and varied and not of a routine mental or physical character. It requires the exercise of original thought and judgment and the ability to supervise the technical and administrative work of others. His/her education will have been such as to make him/her capable of closely and continuously following progress in his/her branch of engineering science by consulting newly published works on a worldwide basis, assimilating such information and applying it independently. He/she is thus placed in a position to make contributions to the development of

engineering science or its applications. His/her education and training will have been such that he/she will have acquired a broad and general appreciation of the engineering sciences as well as thorough insight into the special features of his/her own branch. In due time he/she will be able to give authoritative technical advice and to assume responsibility for the direction of important tasks in his/her branch.

PO7: Environment and sustainability: Understand the impact of the professional engineering solutions in and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

Sustainability is the ability to continue a defined behavior indefinitely. Sometimes environmental, social and economic are termed to be the three pillars of sustainability.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice

The ethics of engineers and the fundamental principles for Engineers are as follows.

Engineers uphold and advance the integrity, honor, and dignity of the engineering profession by:

- I. using their knowledge and skill for the enhancement of human welfare;
- II. being honest and impartial, and servicing with fidelity the public, their employers and clients;
- III. Striving to increase the competence and prestige of the engineering profession; and
- IV. Supporting the professional and technical societies of their disciplines.

PO9. Individual and team work: Function effectively as an individual and as a member or leader in diverse teams, and in multidisciplinary settings.

To work successfully in the team a team member must have the following capabilities.

1. The Ability to Listen

it is important to listen to one another's ideas. Too often in a business setting, you have a group of people simply waiting for their turn to speak, not paying one iota of attention to the persons on their left or right. So it is a good teamwork skill to have the ability to listen

2. Check Your Ego

This isn't saying abandon your ego altogether, because that isn't healthy. But leaving your ego at the door temporarily is a very important team work skill. The reason this is so essential is that there is always someone better than you at something, no matter how brilliant you are.

3. Critique

By critique, I mean constructive criticism. Be able to give others constructive criticism and be able to listen to others critique your ideas and work. There shouldn't be any offense taken to constructive criticism. You all want to succeed, and this is a vital step in doing so.

4. Delegation

The mentality must be applied to teamwork. Delegate roles to those who do them best.

5. Show Respect

If you and another person happen to be paired up and can't stand each other, you can still put that aside for a couple of hours, treat each other civilly, and complete the tasks at hand. You may even overcome the dislike toward one another.

6. Be Helpful

This is simple. If one of your teammates does not understand an idea, discussion, or task that is being completed, take the necessary time to explain it to them and work with them. There are no weak links when everyone helps one another. Some take longer to learn than others, but that doesn't mean that they are of less intelligence. If in a meeting someone asks a question because they don't understand, don't frown at them. Just answer the questions patiently and concisely.

7. Question One Another

If someone brings up a topic of discussion and a solution to this topic, question them. Respectfully question, don't badger. Rather, ask them how it will work, why it will work over the long-run, and how everyone else can implement the idea.

8. Participation

Have the entire team encourage shy people to engage in the topics of discussion. Don't demand it, but make them realize that you really want to hear their ideas.

9. Rational Debate

Bad ideas are bad for teams. Spirited, friendly, rational debate is where facts come forward, ideas are born, and quality rises to the top.

10. Set The Right Environment

Try to make the space in which your team is assembled as comfortable, relaxing, and inviting as possible. You do not want your team to be tense and with frayed nerves.

PO 10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: Project management and finance: Demonstrate knowledge and understanding of the engineering management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

Project management is the application of processes, methods, knowledge, skills, and experience to achieve the project objectives. In general, project is a unique, transient endeavor, undertaken to achieve planned objectives, which could be defined in terms of outputs, outcomes or benefits.

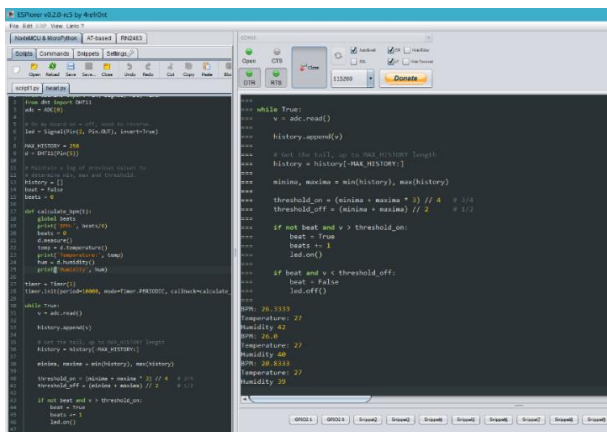
PO12: Life-long learning: Recognize the need for and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.

Life Long Learning means is the provision or use of both formal and informal learning opportunities throughout people's lives in order to foster the continuous development and improvement of the knowledge and skills needed for employment and personal fulfillment

CHAPTER 4

PROJECT IMPLEMENTATION

The system that we proposed is successfully prepared and tested. The patient who will be wearing the device will get his/her heart rate, body humidity & temperature on the IP address of the NodeMCU used using the ESP8266 module. The data can be seen by the doctor and the user itself on the website. To use this system both user and doctor have to register and login into the system. The data of the user can be further used by the doctor to monitor and do an analysis of the patient's health for future discrepancies.



```
from dht import DHT11
import time
import sys
import json
import urllib2

# Pin definitions
DHT11_PIN = D1
DHT11_TEMP = D0

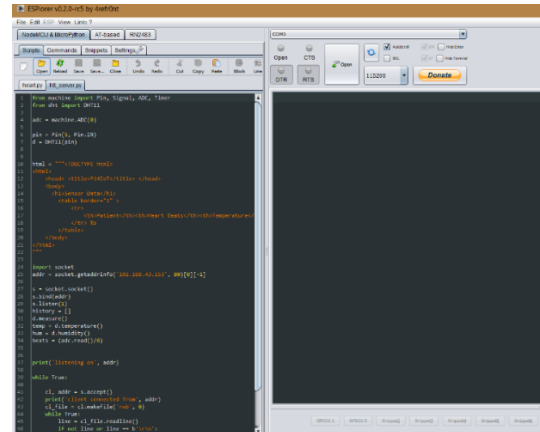
# Initialize DHT11 sensor
dht = DHT11(DHT11_PIN, DHT11_TEMP)

# Variables
history = []
minima = 0
maxima = 0
beats = 0

def calculate_bpm():
    global beats
    beats += 1
    # Read temperature and humidity
    temp, hum = dht.read()
    print("Temperature: %d, Humidity: %d" % (temp, hum))
    # Post data to server
    url = "http://192.168.1.100:8080/postData"
    data = {"temp": temp, "hum": hum}
    req = urllib2.Request(url, data)
    response = urllib2.urlopen(req)
    result = response.read()

# Main loop
while True:
    # Read data from DHT11
    temp, hum = dht.read()
    # Append to history
    history.append((temp, hum))
    # Calculate minima and maxima
    minima = min(history)
    maxima = max(history)
    # Calculate threshold
    threshold_on = (minima + maxima) // 2
    threshold_off = (minima + maxima) // 2
    # Check if beat and v > threshold_on
    if not beat and v > threshold_on:
        beat = True
        beats += 1
        bpm.on()
    # If beat and v < threshold_off
    if beat and v < threshold_off:
        beat = False
        bpm.off()
    # Print BPM
    print("BPM: %d" % bpm.get_bpm())
    # Print temperature and humidity
    print("Temperature: %d, Humidity: %d" % (temp, hum))
    # Post data to server
    calculate_bpm()
    # Sleep for 10 seconds
    time.sleep(10)
```

Figure.8 Micropython Code I



```
from machine import SPI, SDCARD, SDCARD_SIZE
import time
import sys
import json
import urllib2

# Pin definitions
SDCARD_PIN = D1
SDCARD_TEMP = D0

# Initialize SDCARD
sdcard = SDCARD(SDCARD_PIN, SDCARD_TEMP)

# Variables
history = []
minima = 0
maxima = 0
beats = 0

def calculate_bpm():
    global beats
    beats += 1
    # Read temperature and humidity
    temp, hum = sdcard.read()
    print("Temperature: %d, Humidity: %d" % (temp, hum))
    # Post data to server
    url = "http://192.168.1.100:8080/postData"
    data = {"temp": temp, "hum": hum}
    req = urllib2.Request(url, data)
    response = urllib2.urlopen(req)
    result = response.read()

# Main loop
while True:
    # Read data from SDCARD
    temp, hum = sdcard.read()
    # Append to history
    history.append((temp, hum))
    # Calculate minima and maxima
    minima = min(history)
    maxima = max(history)
    # Calculate threshold
    threshold_on = (minima + maxima) // 2
    threshold_off = (minima + maxima) // 2
    # Check if beat and v > threshold_on
    if not beat and v > threshold_on:
        beat = True
        beats += 1
        bpm.on()
    # If beat and v < threshold_off
    if beat and v < threshold_off:
        beat = False
        bpm.off()
    # Print BPM
    print("BPM: %d" % bpm.get_bpm())
    # Print temperature and humidity
    print("Temperature: %d, Humidity: %d" % (temp, hum))
    # Post data to server
    calculate_bpm()
    # Sleep for 10 seconds
    time.sleep(10)
```

Figure.9 Micropython Code II

Figure.8 & 9 shows the programming code based on Micropython for NodeMCU which consist of the libraries and method to retrieve data from sensors and post on the server of NodeMCU.

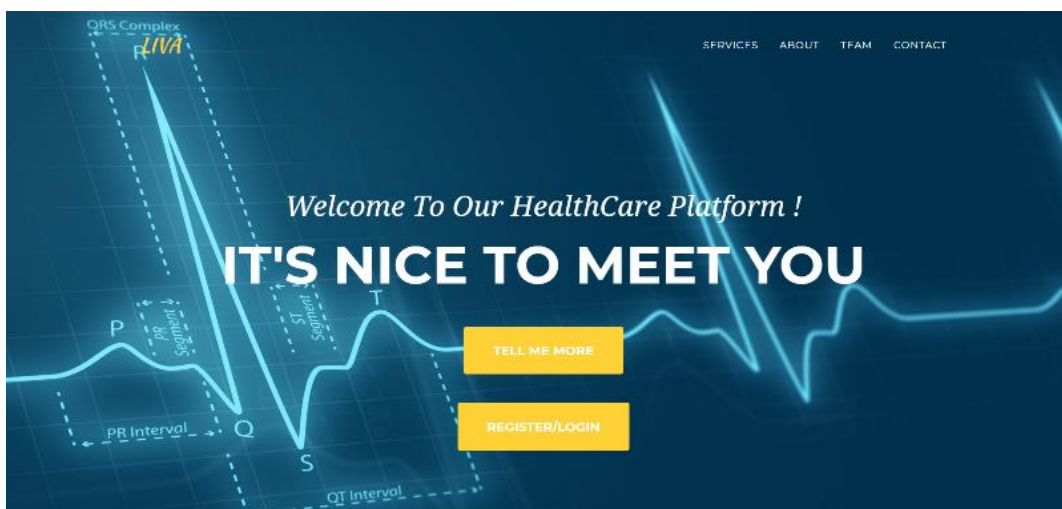


Figure.10 Home Page

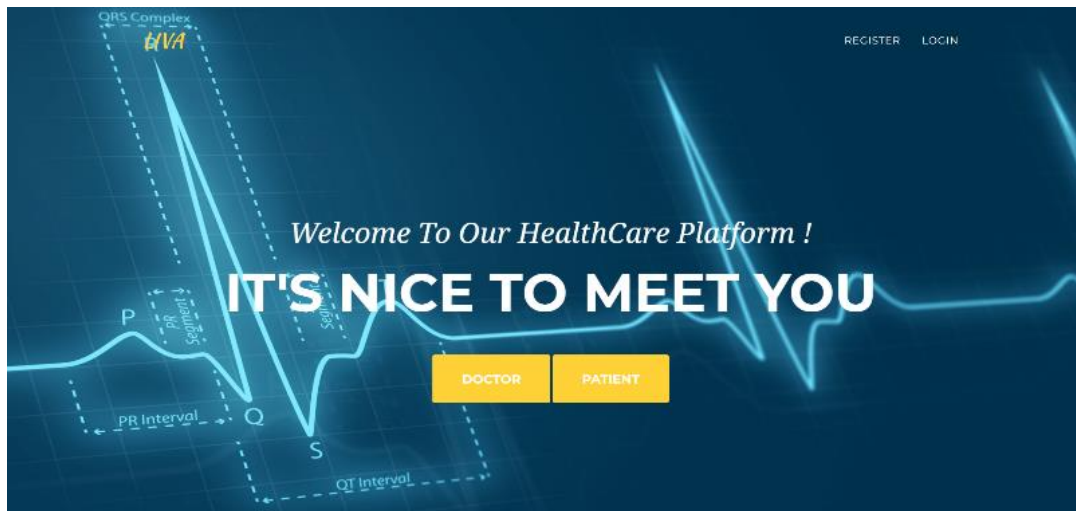
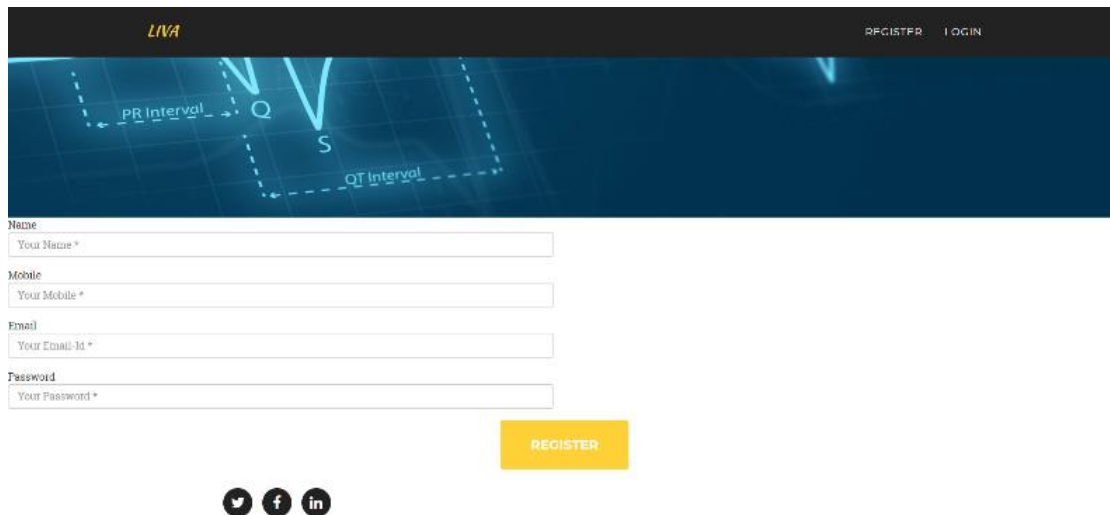


Figure.11 Register page

Figure.10 shows the Home page of the website which comprises of services, about, team and contact. The services tab will describe the services provided by the system, the about tab will describe the whole system and contact tab comprises of the contact form. Figure.11 shows that there are different pages for registration of patient and doctor and can be seen below.

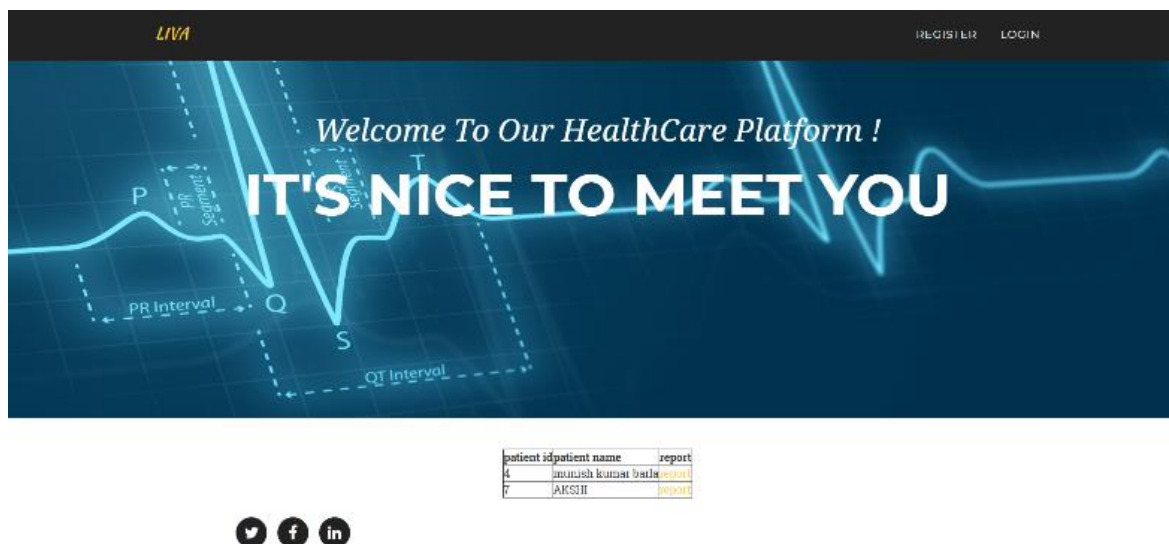
LIVA REGISTER LOGIN
 Name
 Your Name *
 Age
 Your Age *
 Address
 Your Address *
 Mobile
 Your Mobile *
 Email
 Your Email-id *
 Doctor
 Ishan Verma
 Password
 Your Password *
 REGISTER
 Twitter Facebook LinkedIn

Figure.12 Patient Registration Page



The image shows a web page for doctor registration. At the top, there is a dark header with the 'LIVA' logo on the left and 'REGISTER' and 'LOGIN' links on the right. Below the header is a large banner featuring a blue background with a white ECG line. The ECG line has labels for 'PR Interval', 'Q', 'S', and 'QT Interval'. Below the banner, there is a registration form with the following fields: 'Name' (with a sub-label 'Your Name *'), 'Mobile' (with a sub-label 'Your Mobile *'), 'Email' (with a sub-label 'Your Email-Id *'), and 'Password' (with a sub-label 'Your Password *'). To the right of the form is a yellow 'REGISTER' button. Below the form and button are three social media icons: Twitter, Facebook, and LinkedIn.

Figure.13 Doctor Registration page



The image shows a web page for a patient list. At the top, there is a dark header with the 'LIVA' logo on the left and 'REGISTER' and 'LOGIN' links on the right. Below the header is a large banner featuring a blue background with a white ECG line. The ECG line has labels for 'P', 'PR Interval', 'Q', 'S', 'T', and 'QT Interval'. The banner also contains the text 'Welcome To Our HealthCare Platform !' and 'IT'S NICE TO MEET YOU'. Below the banner, there is a table with the following data:

patient id	patient name	report
4	manish kumar badi	report
7	AKSHII	report

Below the table are three social media icons: Twitter, Facebook, and LinkedIn.

Figure.14 Patient List

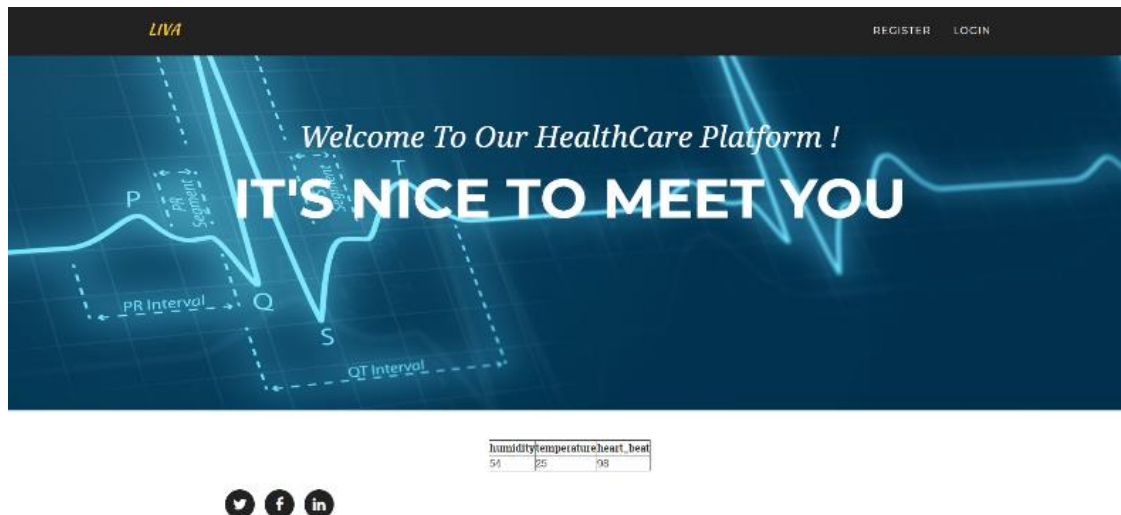


Figure.15 Patient Data

Figure.14 shows the list of patients assigned to the particular doctor who has logged into the system. Figure.15 shows the data of the particular patient selected by the doctor from his/her list or the data can be seen by the patient itself when he/she will log in to the system.

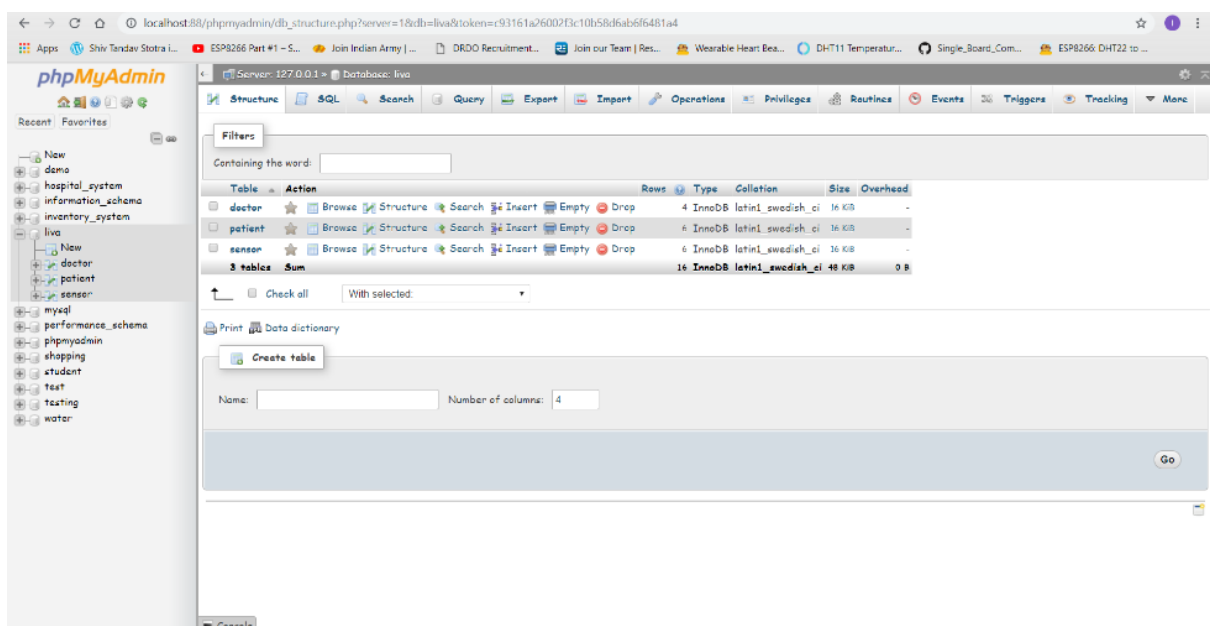


Figure.16 MySQL Database

Figure.16 shows the database designed for the system, the database used is MySQL. The database comprises of three tables - one for data of doctors named “doctor”, second for the data of patient which also comprises of the doctor id who is assigned to that patient named “patient”, and third for the data of patients coming from the device.

CONCLUSION

5.1 Results

The proposed outcome of the project is to give proper and efficient medical services to patients by connecting and collecting data information through health status monitors which would include patient's heart rate, blood pressure, and ECG and sends an emergency alert to patient's doctor with his current status and full medical information.

5.2 Conclusion

The main idea of the proposed system is to provide better and efficient health services to the patients by implementing a networked information cloud so that the experts and doctors could make use of this data and provide a fast and an efficient solution.

The final model will be well equipped with the features where a doctor can examine his patient from anywhere and anytime. Emergency scenario to send an emergency mail or message to the doctor with the patient's current status and full medical information can also be worked on.

The proposed model can also be deployed as a mobile app so that the model becomes more mobile and easier to access anywhere across the globe.

5.3 Future Scope

The IoT can redefine how healthcare is provided to the end customers. The development of smart, wearable devices such as the Apple Watch can make healthcare more accurate, precise and timely. What such devices do is to provide accurate and real-time information to healthcare providers to take action, if needed.

Intelligent Implantable Medical Devices

Medical device makers such as those of pacemakers can implant intelligent sensors on pacemakers that can relay important information on the health of the person wearing it, the makers can receive status on the functioning of the pacemaker and can alert the customer if the pacemaker needs to be repaired or replaced.

Intelligent Personal Medical Devices

Smart devices are going to bring a lot of benefits, such as patients not needing to visit a clinic for monitoring basic vitals such as blood pressure, weight, pulse, and heart rate. The devices are going to relay the information to the doctor and the doctor can prescribe medication if needed. Smart devices will not only relay important health parameters but also provide intelligent alerts to the person who is wearing it. For example, the Lumbo Lift device which measures your posture can send you an alert when you are slouching or you need to correct your posture.

Instant Information

Intelligent devices can provide instant feedback on your health requirements. For example, when you are exhausted after a hard day at work, your watch could measure your vital parameters and give customized recommendations on diet and the kind of relaxation exercises you might need.

ANNEXURES

References

<https://www.wikipedia.org/>

<https://www.w3schools.com/>

<https://roboindia.com/tutorials/DHT11-NodeMCU-arduino>

<https://www.instructables.com/id/ESP8266-DHT22-to-MYSQL/>

<https://micropython.org/>

<http://docs.micropython.org/en/v1.9.4/esp8266/esp8266/tutorial/intro.html/>

Research Paper

[1] Bhoomika.B.K, Dr. K N Muralidhara “Secured Smart Healthcare Monitoring System Based on IoT” 2015 International Journal on Recent and Innovation Trends in Computing and Communication Volume:3 Issue:7.

[2] Ms.Vaishali.G.Hadole, Ms. Durga S Ingle, Mr.Vaibhav.A.Hingne and Prof. Manjusha M. Patil “IOT Based Smart Healthcare kit” 2017 International Journal of Trend in Research and Development, Volume 4(5).

[3] K. Dinesh, K. Vijayalakshmi, C. Nirosha, I. Siva Rama Krishna “IoT Based Smart Health Care Monitoring System” 2018 K. Dinesh et al. International Journal of Institutional & Industrial Research ISSN: 2456-1274, Vol. 3, Issue 1.

[4] Shubham Banka, Isha Madan and S.S. Saranya “Smart Healthcare Monitoring using IoT” 2018 International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 15

[5] Punit Gupta, Deepika Agrawal, Jasmeet Chhabra, Pulkit Kumar Dhir “IoT based smart healthcare kit” 2016 2016 International Conference on Computational Techniques in Information and Communication Technologies (ICCTICT)