LONDON METROPOLITAN UNIVERSITY

islington college
(इस्लिङ्टन कलेज)

Module Code & Module Title

CS6P05NI Final Year Project

ECG Monitoring and its Classification

Assessment Type

40% Final Report

Year and Semester

2020-21 Autumn

Student Name: Sandesh Chhetri

London Met ID: 18029730

College ID:np01nt4a180052

Assignment Due Date: 26 April, 2021

Assignment Submission Date: 26 April, 2021

Word Count: 7400

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

Abstract

The use of the Network can be seen in people's everyday behaviour in the age of modern technologies. People have begun to worry about their wellbeing and issues that can affect their health, according to the health awareness habits observed. Since any human's heart rate is so vital, it should be checked on a daily basis to keep on the safe side.

This document provides the reader with a basic understanding of ECG, its present state in Nepal, issues associated with heart beat testing, and how this project contributes in their treatment. Related programs have been thoroughly examined in order to achieve the listed objective. The latest issue that motivates people to bring features to this project. An appropriate method for the project will be chosen, and tasks will be completed based on the methodology for the project's progress, which is outlined in this document. The technical terminology, programming languages, and frameworks that will be used in the development of this project are also discussed. As a result, this document includes the requirements gathering, specification, execution, and future plan stages.

Table of Contents

Table of Figures

Table of Tables

Chapter 1: Introduction

1.1 Project Description

The most common causes of death are considered to be cardiovascular diseases (CVD). A rising number of deaths caused by chronic and cardiovascular diseases (CVDs) have occurred in all countries around the world in the last decade. The experimental findings on ECGs from actual patient datasets indicate that the average detection rate is 96.1 with an average false alarm rate (FPR) of 1.3. The derived features in the ECG signal would be entered into the classifier to diagnose heart disease. It is important for heart disease clinical diagnosis. We aim to identify Heart Arrhythmia based on the ECG of a patient in this project. With the help of ECG diagram and the software This project will let them know about their heart beat and what problems or is there any arrhythmias or not. Now people don't have to go to hospital for that people with normal understanding also can easily perform the test their ECG.

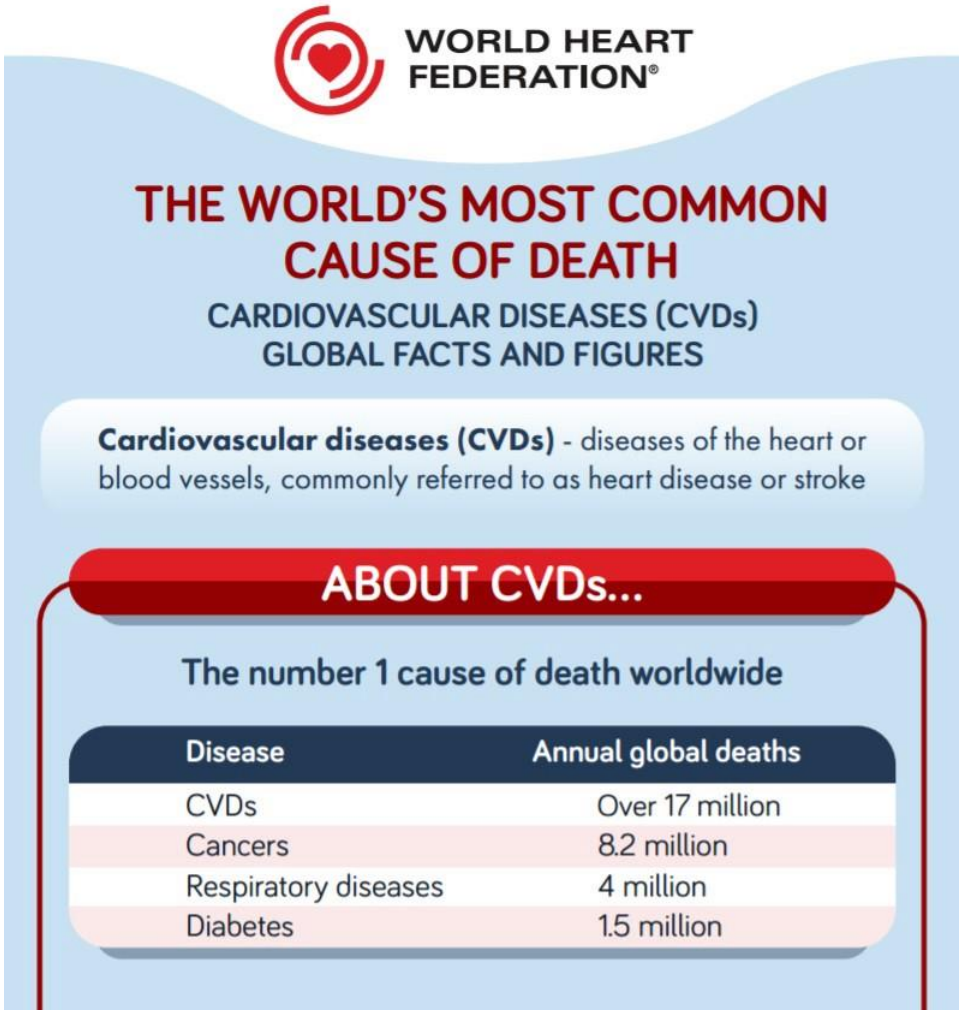

Figure 1 Common cause of death (Singh, July, 2018)

1.2 Current Scenario

The outcome of this project will the device to see the ECG diagram and a software which analyses the diagram and say whether the heart beat is normal or not. Also, which disease is the person likely to suffer from. It is going to be completed with own algorithm. The data are about 93% to 94% accurate but my target is that my proposed project will be about 96% to 97% accurate. People are less aware and not concerned about their heath and some are lazy to go to hospital so those people can easily use this project and check their heart condition and allow them to take necessary precautions that can recover their heart rates.

In addition, the proposed ECG arrhythmia classification method can be applied to the medical robot or scanner that can monitor the ECG signals and helps medical experts to more accurately and easily identify ECG arrhythmia.

1.3 Problem Scenario and Solution

Problem Scenario

When the heart beats too quickly, too slow, or irregularly, a disturbed heart rhythm is This is referred to as arrhythmia. The experimental findings on ECGs from actual patient datasets indicate that the average detection rate is 96.1 with an average false alarm rate (FPR) of 1.3 (Moody & Mark, 2001). Electrodes are used to capture different cardiac signals from the patient. There will be possibility of arrhythmias cause clinical problems or are life threatening. Although you should mention some irregular heart rate to the doctor to be on the safe side. For that people need to visit to doctor in a time interval but most of people don't have any time or care for that and they will get heart disease which can also take your life.

There is also no proper education for people regarding the danger of arrhythmias. Normal heart beat people can also get this suddenly due to many reasons like low potassium and other electrolytes, health condition taking drugs and all.

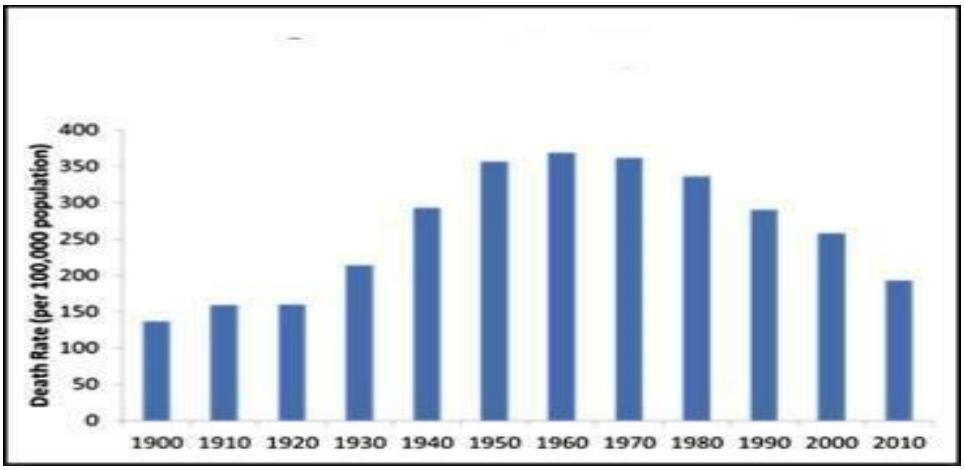

Figure 2: Heart Disease deaths rate (Dalen, et al., 2014)

Project Solution

With the help of ECG diagram and the software This project will let them know about their heart beat and what problems or is there any arrhythmias or not. Now people don't have to go to hospital for that people with normal understanding also can easily perform the test their ECG. The most valuable thing for people is time so this project will also reduce time because now people don't need to visit hospitals frequently to check their heart.

Using my project will be very time saving because in short period of time we can measure and see if we are having arrhythmias and is there chance of any heart disease or not. If we get to know that in time, we can follow some solution like taking rest, keeping your mind calm or even go to hospital in time because if we are not aware of our heart condition then we know the heart disease can surprising attack you and even our life may be in risk.

1.4 Aims and Objectives

Aims

The main aim of this project is to let people know about their heart condition weather the ECG is normal or not and If not, then which disease are they likely to have and take necessary precautions in time which save people life.

Objectives

- To make people aware about their heart condition.
- To use machine learning for making this project.
- To make complete software in given time period which can be used by people which could even save their life.
- To make is less time consuming.
- To analyse the ECG and show the condition of the heart.
- To make it user friendly so anyone with basic knowledge can use.

1.5 Report Structure

1.5.1 Background

The background provides a deeper overview of the process by clarifying the criteria and overview of the project and its expected outcomes. It also helps to learn from similar projects and link them to similar projects. Projects to provide a deeper understanding of the characteristics and elements.

1.5.2 Development

Development clarifies how the project will be built. This explains the methodologies are considered and chosen and the multiple phases of the selected methodologies are evaluated. It also illustrates pre and post survey results with designs, diagrams and screenshots of development core features and architecture.

1.5.3 Testing and Analysis

The improvement to date is clarified and examined in this section. It offers the latest progress situation, checks it, and demonstrates progress in growth. This shows the progression in hardware and software development with the circuit diagram. It further explains that, according to the Gantt diagram of the initiative, the project was not behind and offers the action plan to compensate the missing time to finish the project.

1.5.4 Conclusion

Here will be the summary of whole report with introduction, it processes and how it will help normal peoples in their daily life as well.

Chapter 2: Background

2.1 Electrocardiodiagram (ECG) Signal

Electrocardiography is the process of producing an electrocardiogram (ECG or EKG), of the electrical activity of the heart using electrodes placed on the skin. These electrodes detect the small electrical changes that are a consequence of cardiac muscle depolarization followed by repolarization during each cardiac cycle (heartbeat) (G, et al., 2017) .Changes in the normal ECG pattern occur in numerous cardiac abnormalities, including cardiac rhythm disturbances, inadequate coronary artery blood flow and electrolyte disturbances. A typical ECG waveform comprises of an initial P-wave, followed by the main QRS complex and then a trailing T-wave. These waves are defined as follows:

- P-wave: The low voltage fluctuation caused by the depolarization of the atria prior to contraction. The atria contain very little muscle and thus the voltage change is quite small.
- QRS complex: The largest-amplitude portion of the ECG caused by ventricular depolarization. The time during which ventricular contraction occurs is referred to as systole
- T-wave: Caused by ventricular repolarization. Although the R-peak is often the largest amplitude component, the morphology of a healthy ECG can vary greatly from patient 3 Chapter 1 Introduction to patient with the P- or T waves sometimes dominating or merging with the QRS complex.

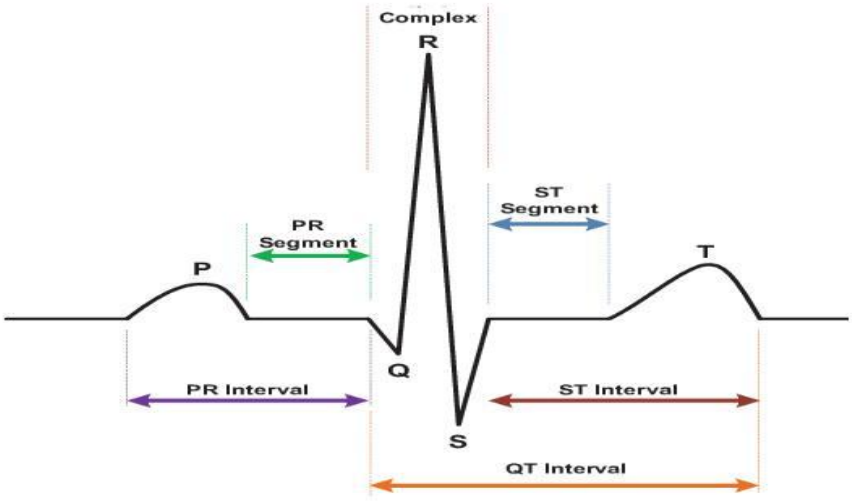

Figure 3 Schematic Diagram of ECG signal (Medina & Farid, 2015)

2.2 Arrhythmia in ECG signal

The normal rhythm of the heart where there is no disease or disorder in the morphology of ECG signal is called Normal sinus rhythm (NSR). The heart rate of NSR is generally characterized by 60 to 100 beats per minute. The regularity of the R-R interval varies slightly with the breathing cycle. When the heart rate increases above 100 beats per minute, the rhythm is known as sinus tachycardia. This is not an arrhythmia but a normal response of the heart which demand for higher blood circulation. If the heart rate is too slow then this is known as bradycardia and this can adversely affect vital organs. When the heart rate is too fast, the ventricles are not completely filled before contraction for which pumping efficiency drops, adversely affecting perfusion.

The different types of arrhythmia that will be used in this project are:

a) **Normal (NOR)**

When you're not doing it, the heart rate should be between 60 and 100 beats per minute. This is what the resting heart rhythm is called. You must wait at least five minutes after being busy before taking your pulse.

Your heart pumps harder when you're busy in order to bring more blood to your functioning muscles. Your heart will pound faster the more the body is working. When sprinting, for example, the heart rate would be significantly higher than when walking. It's natural for your heart rate to reach 160 beats per minute or higher when you're working out hard.

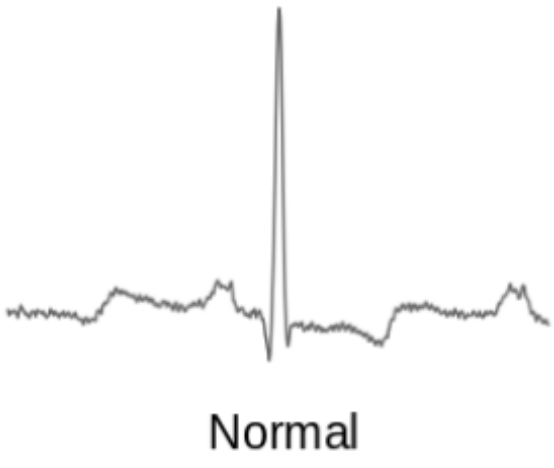

Figure 4 Normal heart beat

b) Premature Ventricular Contraction (PVC)

Extra heartbeats that starts during one of your heart's two lower pumping chambers are known as premature ventricular contractions (PVCs) (ventricles). These extra beats throw off your normal heart rhythm, giving you a fluttering or missed beat sensation in your chest.

Symptoms:

- Chest pounding
- Missed o skipped beats
- Fluttering
- You'll be more conscious of your pulse.

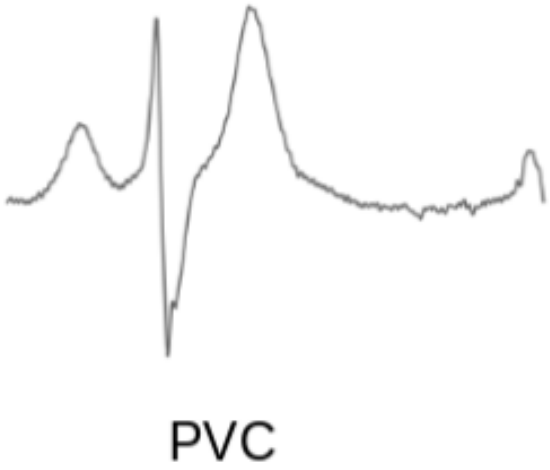

Figure 5 PVC heart beat

c) Paced Beat (PAB)

On a 12-lead ECG, the occurrence of irregularly irregular RR cycles of fibrillatory waves during pacer inhibition or the presence of clear fibrillatory waves and no discernible P waves between pacemaker spikes was used to diagnose AF underlying a pacing rhythm (Patel, et al., 2000).

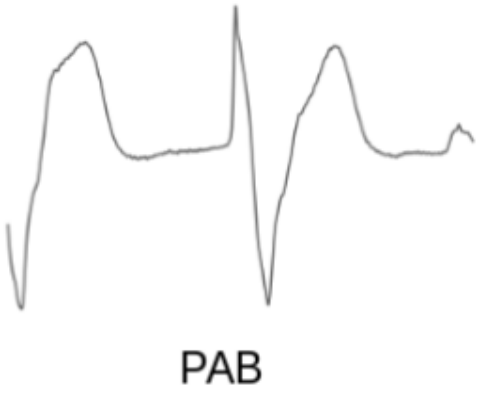

Figure 6 PAB heart beat

d) Right Bundle Branch Block (RBB)

Since the electrical impulse hits, a roadblock at the left or right branch of the "bundle of His," doctors call the resultant electrical pattern bundle branch block. The packet of His is a section of the heart that transmits impulses between the left and right ventricles. A blockage of electrical impulses to the right ventricle of the heart is known as right bundle branch block.

Symptoms:

- Fainting time and again
- Feeling like fainting.

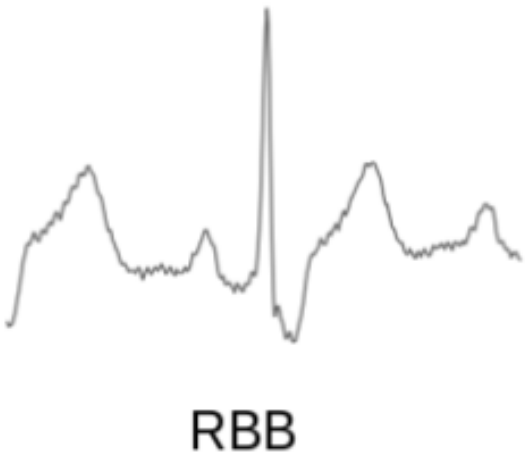

Figure 7 RBB heart beat

e) Left Bundle Branch Block (LBB)

An issue with the electrical wiring (conduction) mechanism of the heart is known as a left bundle branch block. The signal causes nearby areas of the heart to contract in a clockwise motion as it flows through the heart's conduction system. A left bundle branch block is still present in some individuals.

Others experience it on and off, based on their heart rate. A left package branch block can also be incomplete in some people. This could indicate that an individual is suffering from a left bundle branch block. Older individuals are more likely to experience left bundle branch block. In stable young adults, it is very unlikely.

Symptoms:

- Short breath and fatigue.
- High blood pressure
- Heart attack

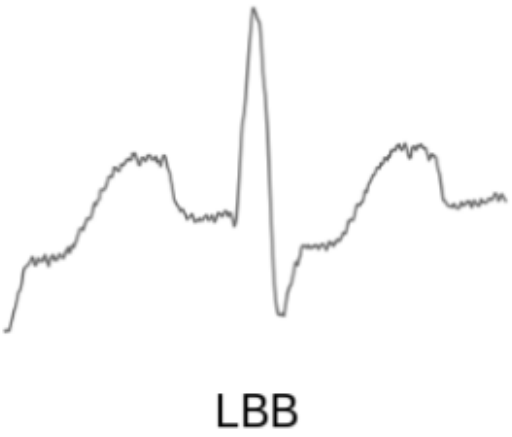

Figure 8 LBB heart beat

f) Atrial Premature Contraction (APC)

A typical form of heart arrhythmia is atrial premature complexes (APCs), which are characterized by premature heartbeats occurring in the atria. Premature atrial contractions are another term for atrial premature complexes. Heart palpitations or an irregular awareness of the heartbeats is one of the most frequent signs of APCs (Selner, September, 2018).

Causes of APC:

- Alcohol
- Stress
- Caffeine
- Poor sleep

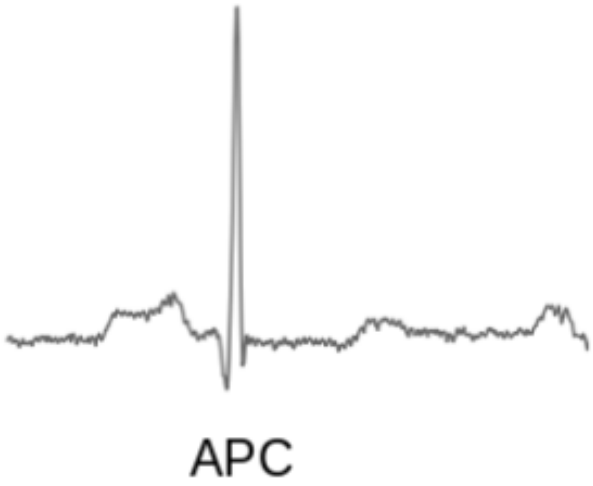

Figure 9 APC heart beat

2.3 Understanding the Project

2.3.1 Project Elaboration

The project is divided in two sections. In first section we will be taking live ECG from a person. In the second section we will be classifying the recorded ECG diagram and detect whether the ECG is healthy or not. We will also be detecting the kind of irregularity in the diagram with the help of our classification algorithm.

Hardware used

a) Single Op-amp (LM741)- The LM741 series of differential amplifier are general-purpose devices that outperform industry expectations. The amplifiers have a number of features that make their use almost fool proof: input and output overload safety, no latch-up when the common-mode range is surpassed, and no oscillations.

b) Five resistors- A resistor is a two-control electrical component that acts as a circuit element by implementing electrical resistance. Resistors are used in electrical circuits for a variety of purposes, including reducing current flow, adjusting signal speeds, dividing voltages, biasing active components, and terminating transmission lines.

c) Battery (9V)- A battery is a device that consists of one or more electrochemical cells connected to the outside system that is used to fuel electrical equipment such

as flashlights, cell phones, and electric vehicles. The positive terminal of a battery is the cathode, and the negative terminal is the anode while it is delivering electric electricity.

d) Breadboard

A breadboard is a popular method for designing and testing circuits. When using a bread board to build a circuit, you don't have to weld wires and parts together. It is simpler to install and reuse modules. Since the connectors are not soldered, you can alter the circuit design at any time.

e) Wires

The means in which energy is transported to and through each personal home that uses electricity is known as electrical cable. It's made of a metal that conducts electricity well, typically copper, and it's encased in a plastic sheath known as an insulator.

Software Used

a) Python

Python is a deciphered, object-arranged, undeniable level programming language with dynamic semantics. Its undeniable level inherent information structures, joined with dynamic composing and dynamic restricting, make it exceptionally appealing for Rapid Application Development, just as for use as a prearranging or paste language to interface existing parts together.

Python's plain, easy-to-learn syntax stresses readability, which lowers software maintenance costs. Modules and packages are supported by Python, which facilitates software modularity and code reuse. The Python parser and its comprehensive standard library are free to download and distribute in source or binary form for all major platforms (Lindstorm, 2005).

b) Spyder

Spyder is an open-source, cross-platform collaborative development environment (IDE) for Python-based science programming. Spyder works for NumPy, SciPy, Matplotlib, pandas, IPython, SymPy, and Cython, as well as other open-source applications. It's under the MIT license. Spyder was originally created and developed by Pierre Raybaut in 2009, but it has been supported and updated by a team of science Python developers and the community since 2012 (Bernard, February, 2018).

2.3.2 About the end users

The project is Low-cost ECG which is mainly targeted to people who cannot afford to go hospital time to time especially targeting people of rural areas. Now people don't have to go to hospital for that people with normal understanding also can easily perform the test their ECG. The most valuable thing for people is time so this project will also reduce time because now people don't need to visit hospitals frequently to check their heart. Using my project will be very time saving because in short period of time we can measure and see if we are having arrhythmias and is there chance of any heart disease or not. If we get to know that in time, we can follow some solution like taking rest, keeping your mind calm or even go to hospital in time because if we are not aware of our heart condition then we know the heart disease can surprising attack you and even our life may be in risk.

2.4 Similar Projects

2.4.1 Project 1: ECG from a single op-amp and 5 resistors.

Author: Scott W Harden

This project aims to create the highest quality ECG machine but rather to create the simplest one with emphasis on predictable and reproducible results. The finished project is a blend of improved hardware and custom cross-platform open-source software (which runs on Windows, Linux, and MacOS), and an impressively good ECG considering the circuit is so simple and runs on a breadboard. He made electrodes by soldering wires to copper pennies, applying power to the penny by super-gluing the cable, and connecting them to his chest using electrical tape (Harden, August, 2016).

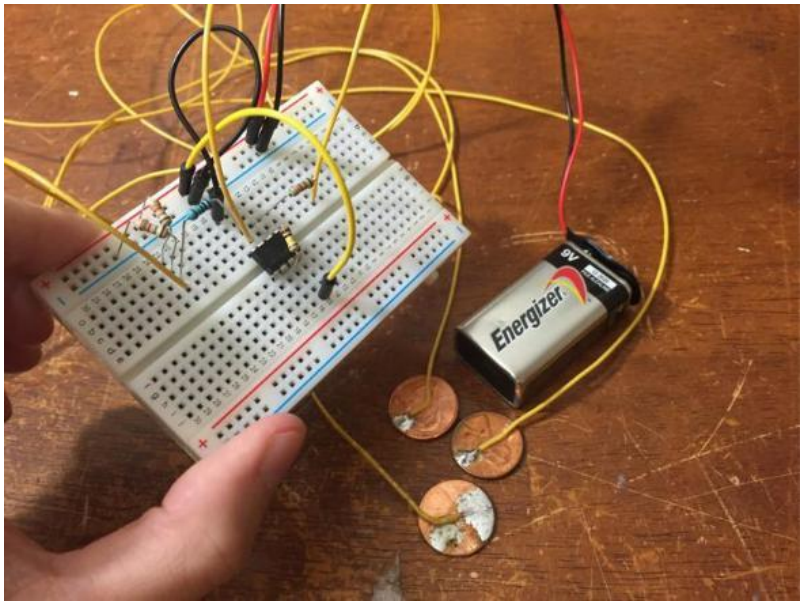

Figure 10 Hardware design

2.4.2 Project 2: ECG arrhythmia classification using a 2-D convolutional neural network

Author: Tae Joon Jun, Hoang Minh Nguyen, Daeyoun Kang, Dohyeun Kim, Daeyoung Kim, Young-Hak Kim

In this article, they suggest an efficient method of classification of electrocardiogram (ECG) arrhythmias using a deep two-dimensional K-nearest neighbors (KNN) that has recently demonstrated excellent success in the field of pattern recognition. With 97.85 percent mean sensitivity, the classifier reached 99.05 percent average precision.

Experimental findings have successfully verified that, without any manual pre-processing of ECG signals such as noise filtering, feature extraction, and feature reduction, the proposed CNN classifier with the transformed ECG images will achieve excellent classification accuracy (Huang, 2017).

2.5 Comparisons of Projects

S. N	Features	Project 1	Project 2	My project
1.	Electrodes used	Copper	X	muscle stimulator pads
2.	Op-amp used	LM324	X	LM741
3.	Classification algorithm used	X	KNN	CNN
4.	Easy to use	Yes	Yes	Yes
5.	Web-app to display output.	Yes	X	Yes

Table 1 Project Comparison table

In the above table we have compared the two projects with my project. We can see that the electrodes used in first project was copper, second project didn't used any electrodes and my project will use muscle stimulator as electrode. The Op-amp used in the first project was LM324, second project didn't used any Op-amp and my project will use LM742 as Op-amp this helps to reduce noise and make our diagram more accurate.

For the classification of algorithm first project didn't had its own algorithm, second one used K-nearest neighbours (KNN) and my project will use convolutional neural network (CNN) which a simple supervised machine learning algorithm which solves both classification and regression problems.

2.6 Conclusion from similar projects

In essence, we can clearly justify that the proposed system provides all the features with low cost and high accuracy. Out of the two projects, the proposed project is convolution of all the features provided in the respective project mentioned above.

Chapter 3: Development

3.1 Considered methodology

3.1.1 Agile methodology

It is a form of technique in which the need and the solution develop. Via the teams' and the client's joint action. The assignment is broken down into by engaging with multiple phrases and continual enhancement and iteration, Stakeholders, with them. The parts separated are considered sprints (Littlefield, February, 2019).This methodology was considered as various tasks that could be carried out dividing them into portions but client interaction and involvement in every sprint are not possible because of which it was not selected.

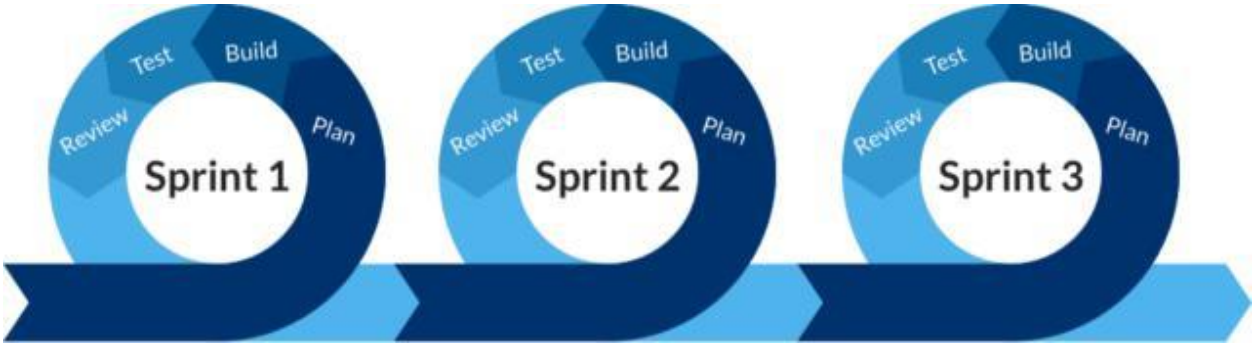

Figure 11 Agile Methodology

3.1.2 Prototype methodology

It is the type of technique under which the original requirement is collected, successive prototypes are produced with added features and improvements and the process is repeated until the product that satisfies the client is produced. The prototype. In the context of this initiative, the most viable approach is methodology (Charvat, 2003).

Types of prototype methodologies-

a) Evolutionary prototype

The prototype is gradually refined in response to consumer reviews until it is fully approved. It allows you to save both time and effort. That's because creating a prototype from scratch for and method interaction can be time-consuming and stressful.

This model is useful when working on a project that involves a new technology that isn't well known. It's also useful for a large project where any feature must be tested once. It's useful because the criterion isn't set in stone or isn't well known at the outset.

b) Rapid throwaway prototype

The criterion for a rapid throwaway is dependent on the preliminary requirement. It's made easily to demonstrate how the requirement would appear visually. The input from the client aids in the modification of the standard, and the version is produced again before the requirement is baselined.

An established prototype will be rejected in this process and will not be used in the final approved prototype. This method is useful for prototyping solutions and receiving immediate input on consumer needs.

c) Extreme prototype

The method of extreme prototyping is most often used in web creation. It is divided into three parts that must be completed in order.

- In HTML format, there is a basic prototype with all of the existing pages.
- A sample services layer can be used to simulate data processing.
- The services are incorporated into the final version and introduced.

d) Incremental prototype

The end product is divided into small prototypes and produced independently in incremental prototyping. The various designs are eventually combined into a single product. This approach aids in reducing the amount of time it takes for the customer to provide input to the application development team.

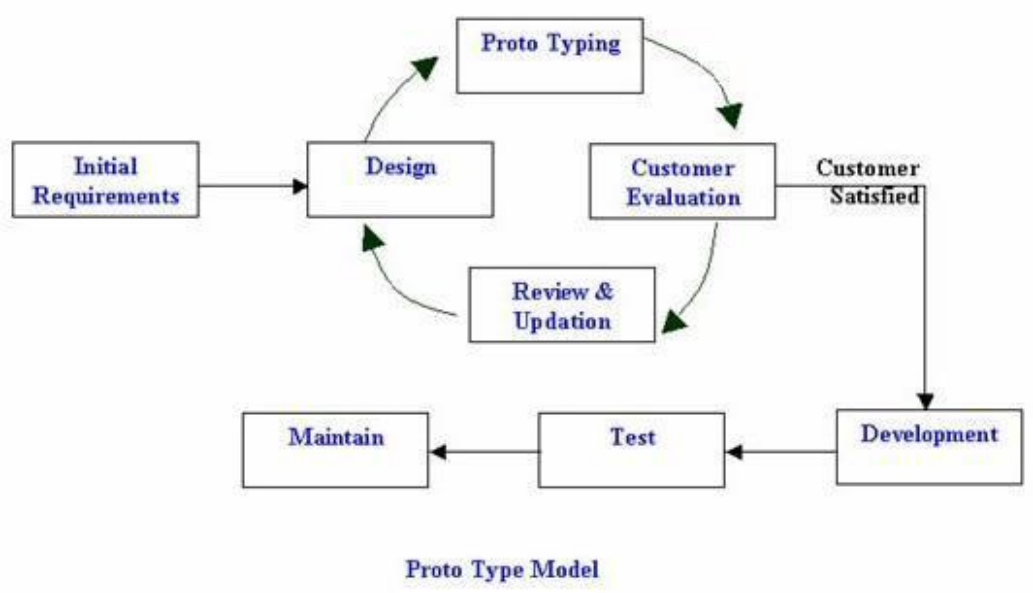

Figure 12 Prototype Methodology (MBA Skool Team, 2018)

3.2 Selected Methodology

For this project I am going to use Waterfall model because a linear sequential flow is the Waterfall Model. In which development is seen as slowly flowing downward (like a waterfall) over the phases of deployment of applications. This suggests that every step in the phase of creation continues only if the previous stage is complete. The waterfall method does not describe the mechanism of managing changes in specifications back to the previous level. The waterfall strategy is the earliest and most widely known approach that has been used for software development.

3.3 Phases of Selected Methodology

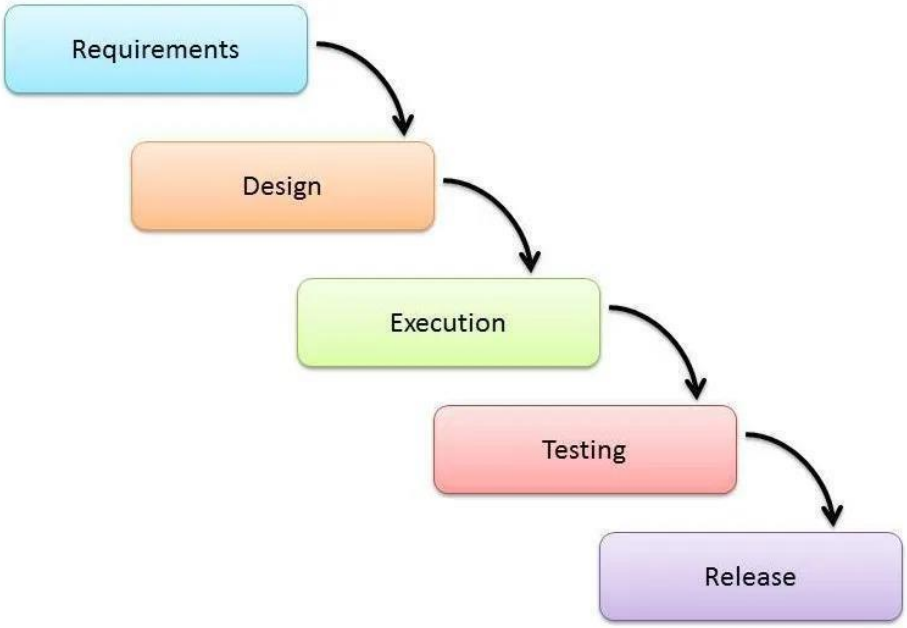

Figure 13 Waterfall model (Sami, March, 2012)

Phases of waterfall method

Requirements

Both project specifications are analysed and recorded in a specification manual during this process, and a feasibility review is performed to see if they are relevant. Any limits or restrictions (e.g., time, budget constraints) that may impact the production process must be considered. A Requirements Understanding Documents (RUD) is developed after thorough analysis.

System Design

In this process, the requirement requirements from the first phase are researched and device architecture is planned. System Design tends to determine specifications for hardware and systems and also helps to define the overall system configuration. The software code that will be written in the next step is now being developed. It is further divided into :

- High-level design phase

  This is the beginning of the device design and architecture phase. It contains a list of modules and their functions, as well as correlations between modules, architecture diagrams, and database tables. The development of a High-level Design document concludes this process.

- Low-label design phase

  This includes the development of actual device modules. The previous phase's High-level Design is disintegrated into multiple components. Each module is defined (pseudo-code) in the Low-level Design manual, allowing the programmer to code directly from the document. It also includes information about each module's interface, error messages, dependency problems, and inputs and outputs.

Execution

The source code is written according to the specifications in this process, as the name implies. The physical design requirements are converted to code. The machine is built up of small programs called modules, which are then linked together. Unit testing is where the configuration of each unit is checked until it is integrated.

Testing

Once each unit has been tested, all of the units created during the deployment process are merged into a system. The program that has been created must be constantly tested to see whether there are any bugs or mistakes. Testing is undertaken to ensure that the customer does not have any issues installing the program.

Release

The program is installed into a live system (the client's server) in this process to be tested. End-users can use the app after it has been implemented. This process can also require real-time user testing in order to convey the system's advantages.

Maintenance

Following implementation, this step includes making changes to the device or a single part in order to change attributes or increase performance. These changes are made in response to consumer demands for changes or to bugs discovered when the product is in operation.

The built software is maintained and supported on a regular basis for the customer (Sharma, April 17, 2016).

Advantages of waterfall model

- The objectives, stages, and tasks are all well identified. As a result, project managers may find it simple to prepare, schedule, allocate money, and identify objectives. Each step has a deliverable, rendering the whole procedure clear and straightforward to understand.
- It's the method of choice for small tasks.
- The entire procedure is carefully recorded and scripted.
- There is less need for reworking since each step is distinct and done within a certain time frame.

Distadvantages of waterfall model

- This design is inflexible; until the phase starts, changes to the specifications are not necessary. As a result, there is no space for self examination or updating.
- This model isn't ideal for large projects with a lot of requirements that change frequently.
- There is no need for criticism during the implementation phase after developers and clients settle on the specifications early in the period.
- It includes a great deal of inconsistent data.
- Testing begins after the implementation process has completed, which may result in the identification of defects and design problems later in the development life cycle.

3.4 Survey Results

This survey was conducted among 10 people of different age group and gender. 7 of the were students and 3 of them were in medical field.

3.4.1 Pre-Survey Results

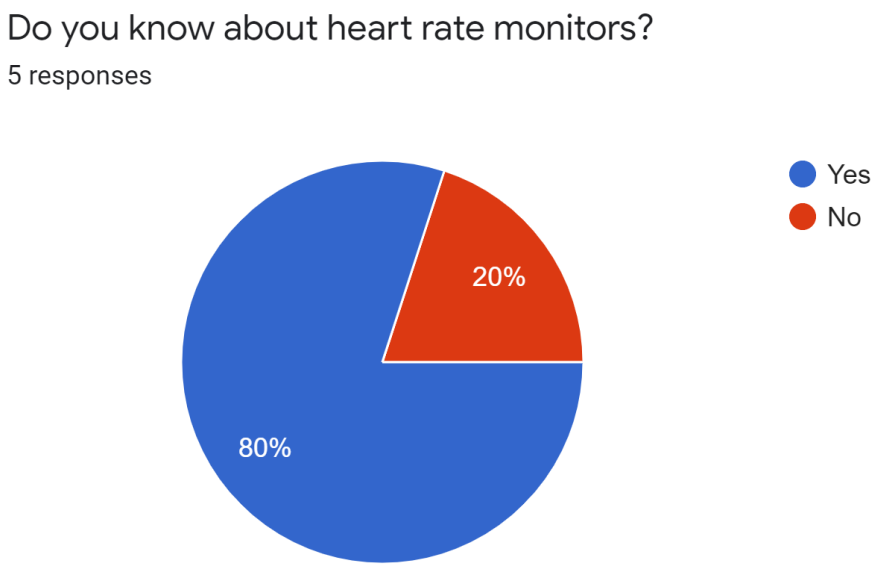

Figure 14 Pre-Survey Results 1

From the above figure 14, 80% of the people know and only 20% of people are unaware about heart rate monitors.

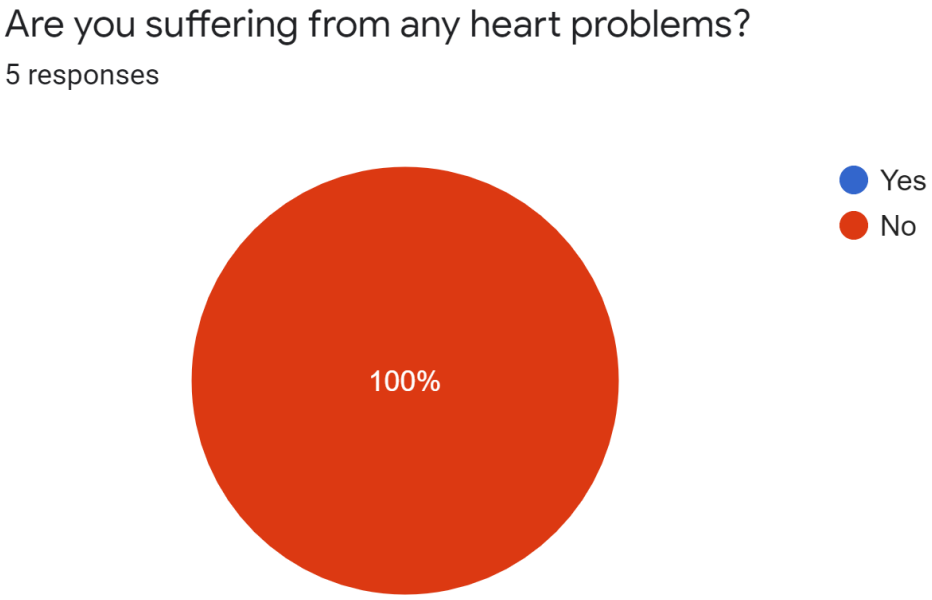

Figure 15 Pre-Survey Results 2

From the above figure 15, we can see that everyone who participated in this survey are healthy.

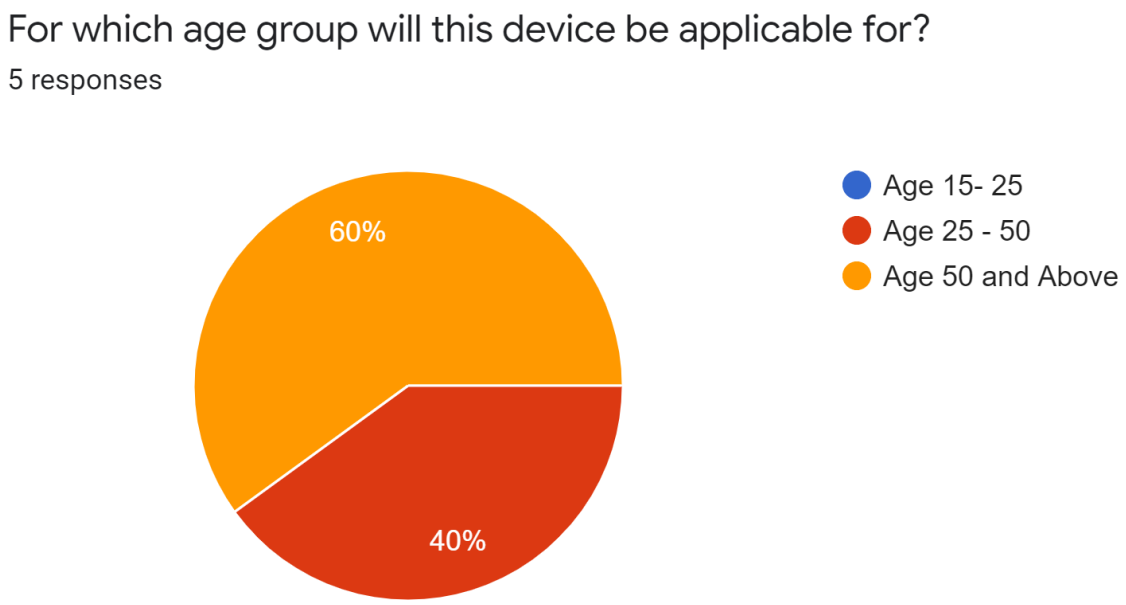

Figure 16 Pre-Survey Results 3

From the above figure16, 60% of the people think that this device is applicable for age group of 50 and Above and 40% of people think that this device is applicable for age group of 25-50.

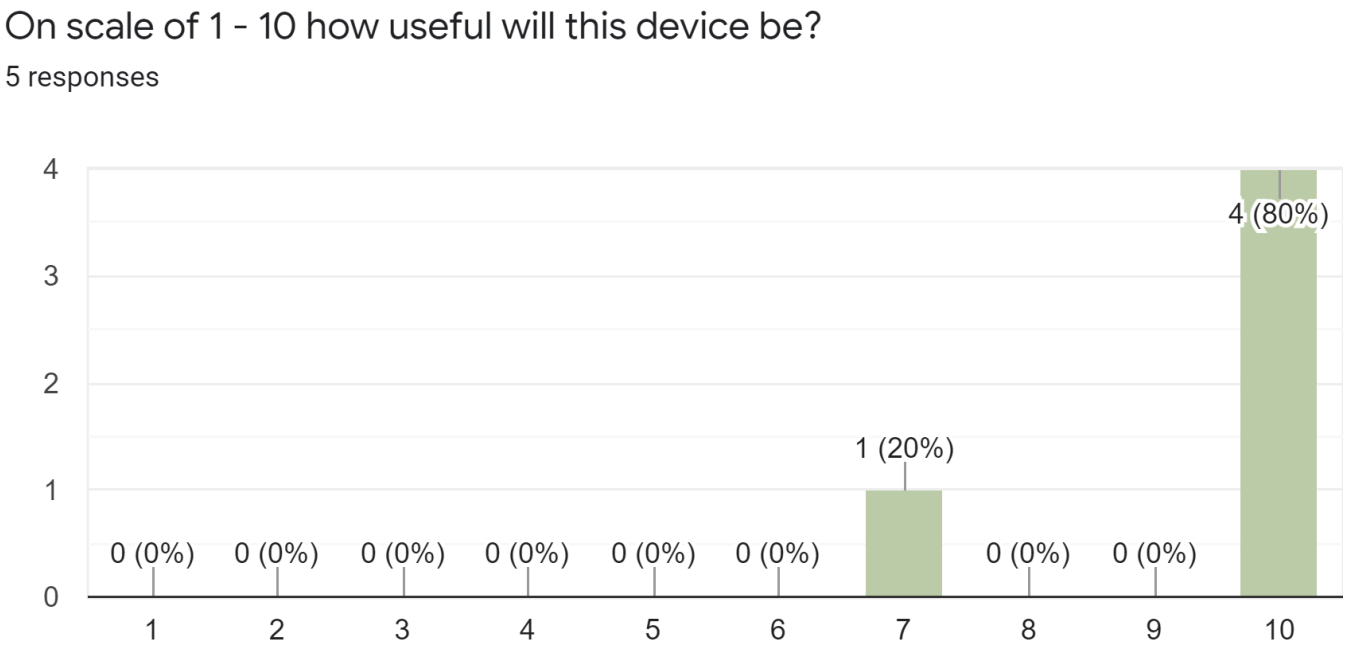

Figure 17 Pre-Survey Results 4

From the above figure 17, we can see that on the scale of 1-10 80% of the people think that this device is useful and remaining 20% think that this device is not useful.

3.4.2 Post-Survey Results

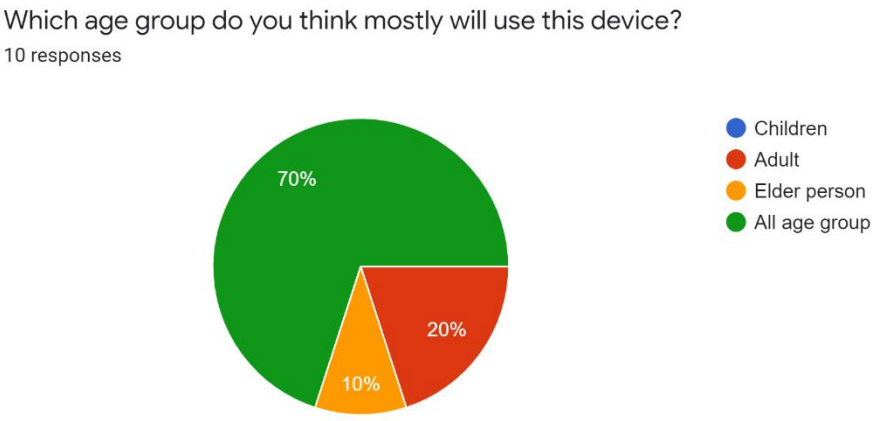

Figure 18 Post-Survey Results 1

From the above figure 18, 70% of the people think that this device will be used by all age group, 20% of people think that this device will be used by adults and 10% of the people think that this device will be mostly used by Elder persons.

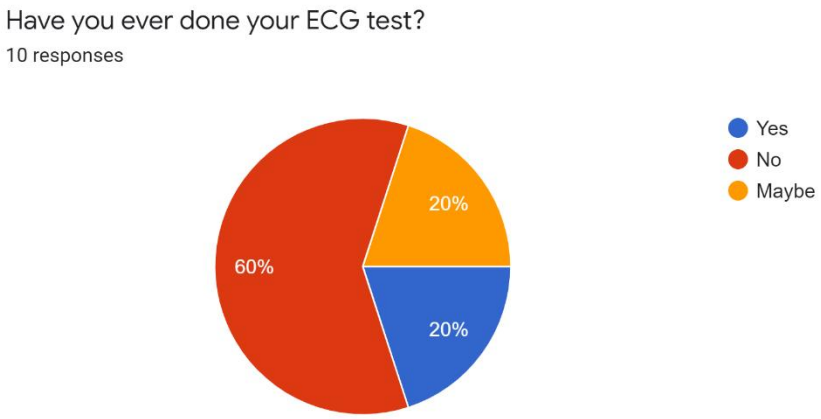

Figure 19 Post-Survey Results 2

From the above figure 19, 60% of the people haven't done their ECG test, 20% of people think that they have done their ECG test and remaining 20% of the people are confuse whether they have done their ECG test or not.

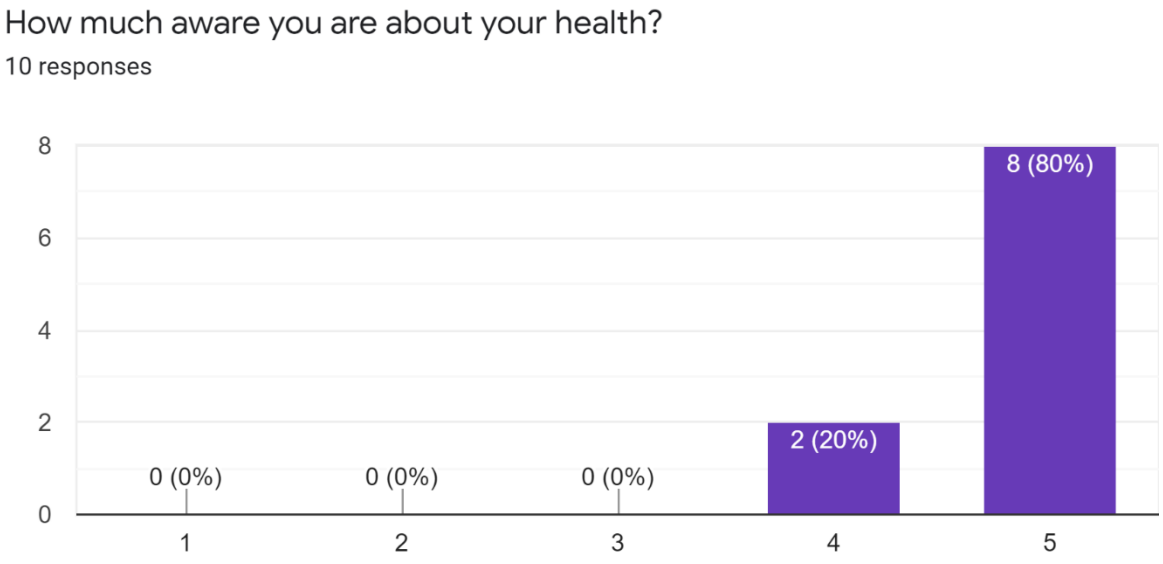

Figure 20 Post-Survey Results 3

From the above figure 20, we can see that on the scale of 1-5 80% of the people are ware about their health and remaining 20% are not aware about their health.

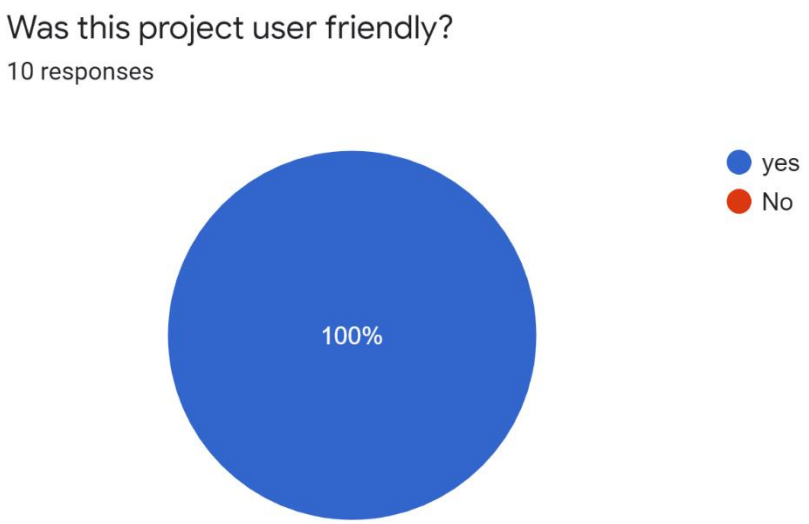

Figure 21 Post-Survey Results 4

From the above figure 21, we can see everyone thinks that this project is user friendly.

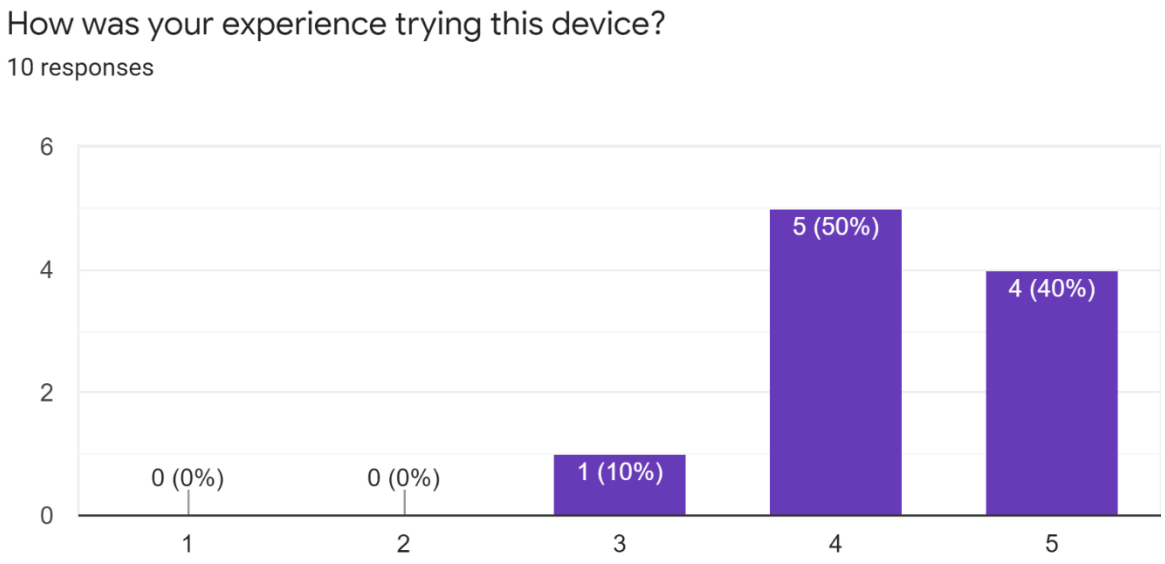

Figure 22 Post-Survey Results 5

From the above figure 22, In the scale of 1-5 where 1 being worst and 5 being excellent. 10% of the people think that their experience while trying this device was good, 50% of people think that their

experience while trying this device was very good and 40% of the people think that their experience while trying this device was excellent.

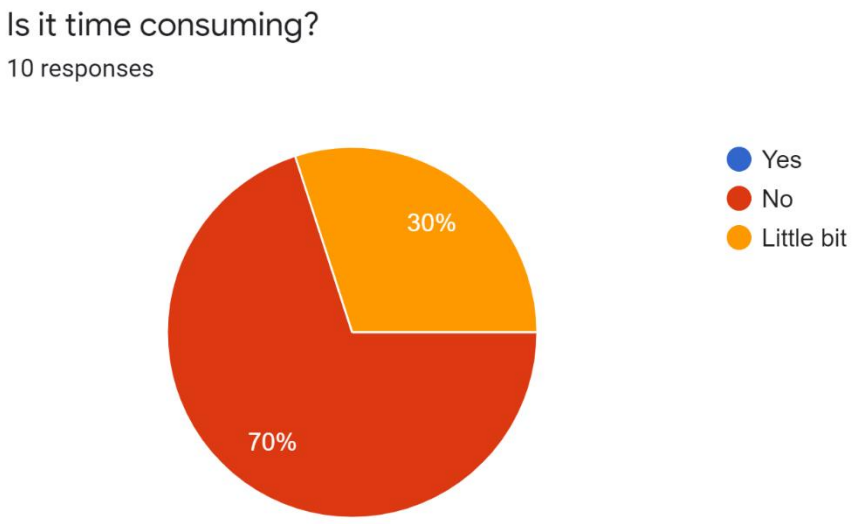

Figure 23 Post-Survey Results 6

From the above figure 23, 70% of the people think that this device is not time consuming at all and 30% of people think that this device is little bit time consuming.

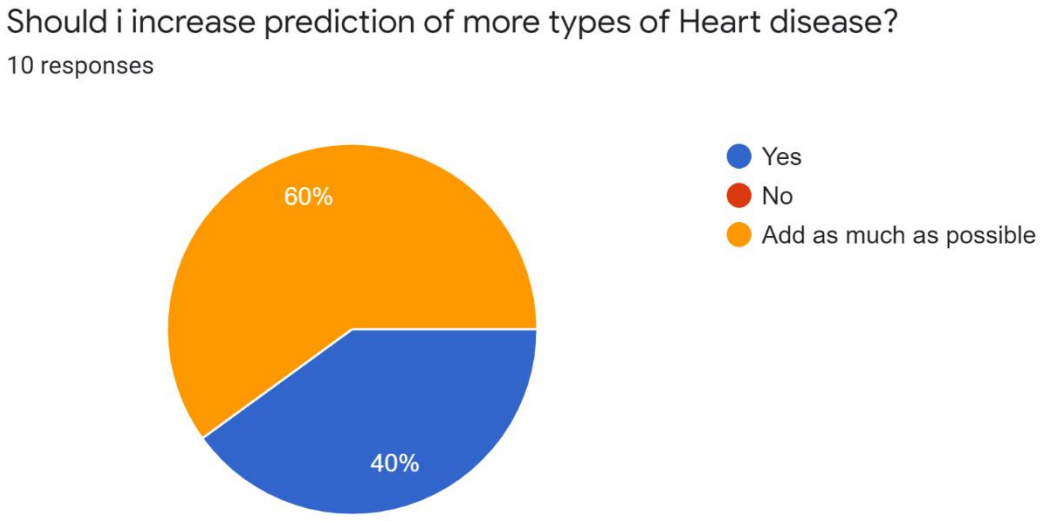

Figure 24 Post-Survey Results 7

From the above figure 24, 60% of the people think that I should increase prediction of heart disease as much as possible and 40% of people think that I should increase prediction of heart disease.

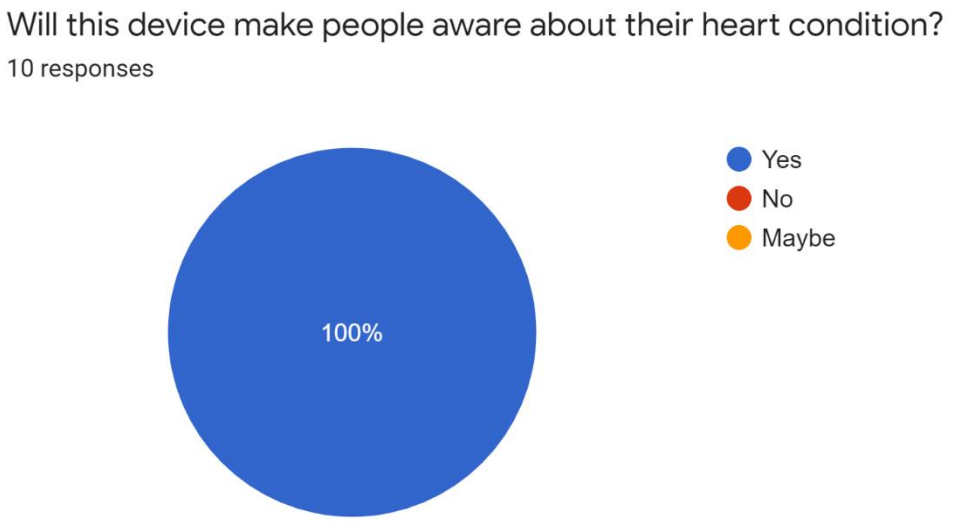

Figure 25 Post-Survey Results 8

From the above figure 25, we can see everyone will be aware about their heart condition after using this device.

The below are the final findings from the post-survey that had an impact on the main project:

- This project will be mostly used by people of all age group.
- Most of the people have not done their ECG test so after testing this device they will do their ECG test.
- Almost everyone was aware about their health.
- Everyone thought that this project was user friendly so anyone can easily use it.
- Most of people felt that the experience of this project was very good so, after upgrading the components used in this device people will have an excellent experience.
- Many of the people think that this device is less time consuming so, anyone could easily be aware about their health in short period of time.
- Currently, this device can only detect limited types of arrythmia. Many of the people have suggested to increase the types of arrythmia as much as possible in the future.

- Everyone think that after using this device, each and every one will be aware about their heart condition. Now the number of people dying from heart disease will reduce by certain number.

3.5 Requirement Analysis

3.5.1 Project Elaboration

The project is divided in two sections. In first section we will be taking live ECG from a person. In the second section we will be classifying the recorded ECG diagram and detect whether the ECG is healthy or not. We will also be detecting the kind of irregularity in the diagram with the help of our classification algorithm.

Hardware used:

- Single Op-amp (LM741)
- Five resistors
- Battery (9V)
- Breadboard
- Wires

Software used:

- Python
- Spyder
- Command Prompt

3.6 Design

Before development of this project diagrams like flowchart and wireframes were created and here are images of those designs:

3.6.1 Flowchart

A flowchart is a schematic description of a series of measures. It is commonly used to present the flow of algorithms, workflows, or procedures because it displays steps in a linear order.

Simple geometric symbols and arrows are used to describe relationships in flowcharts. An oval, for example, represents the start or end of a program in computing. A rectangle is used to represent a process, a diamond is used to represent a judgment, and a parallelogram is used to represent an I/O process. A cloud is used to describe the Internet (TechTarget Contributor, 2008).

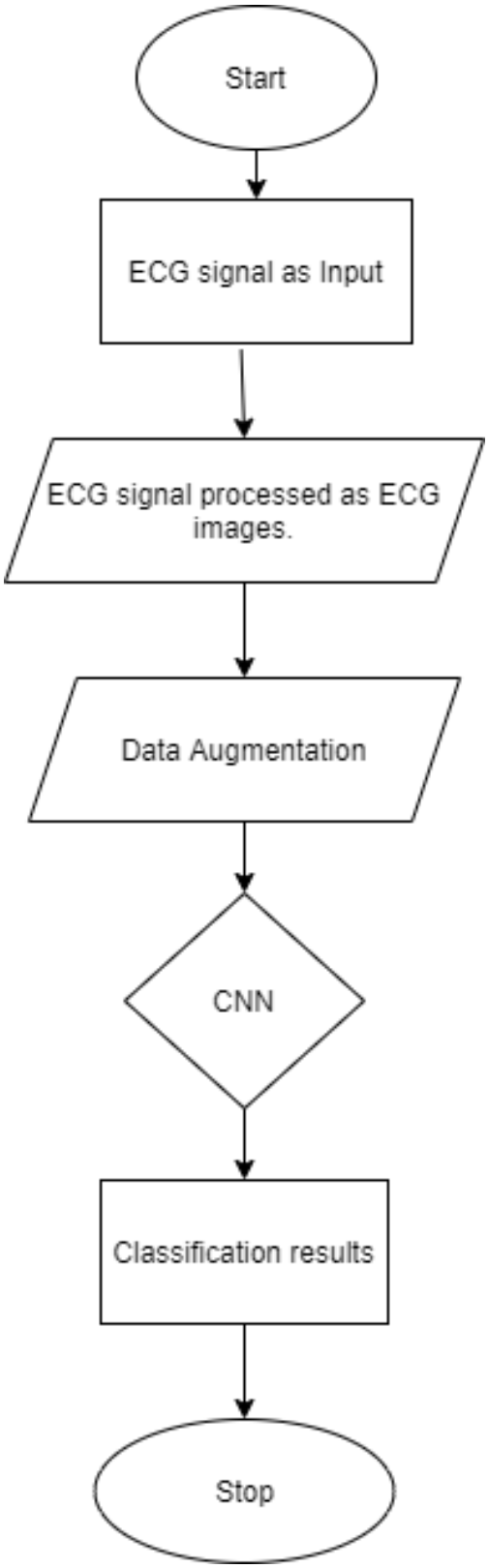

Figure 26 Flowchart

3.6.2 Wireframe

A wireframe is a two-dimensional skeletal outline of a website or app, analogous to an architectural blueprint. Wireframes show the page's configuration, interface, information management, user flow, features, and planned habits in detail. Since a wireframe typically reflects a product's initial design, it's important to keep styling, colour, and graphics to a minimum. Depending on the level of detail needed, wireframes may be drawn by hand or generated digitally (Hannah, April, 2021).

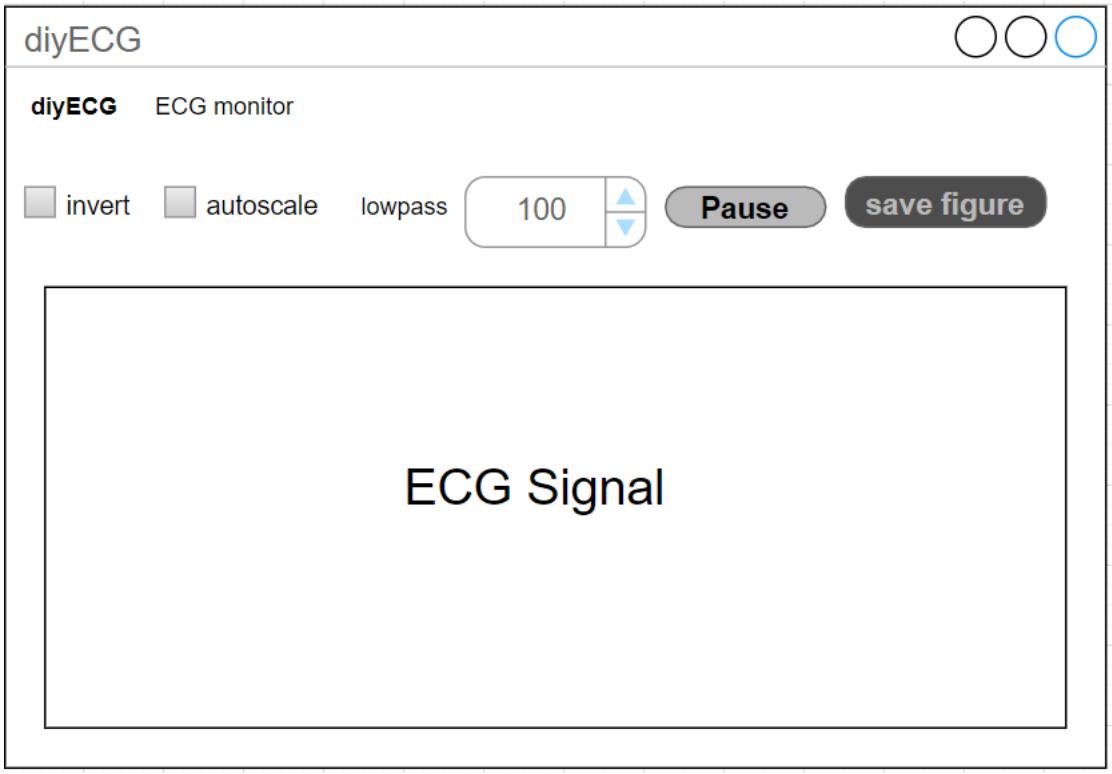

Figure 27 Wireframe of ECG signal

Figure 28 Wireframe of Website 1

Figure 29 Wireframe of Website 2

3.7 Implementation

In this section, important screenshots of the development core feature, architectures are explained with pictures.

3.7.1 Software Implementation

```
def get_records():
    """ Get paths for data in data/mit/ directory """
    # Download if doesn't exist

    # There are 3 files for each record
    # *.atr is one of them
    paths = glob('/path/to/MITDB/dataset/*.atr')

    # Get rid of the extension
    paths = [path[:-4] for path in paths]
    paths.sort()

    return paths

```

Figure 30 Implementation 1

Here the lead2 signals used in this implementation. In the code snippet below, the get records() function generates a list of all the records in the collection.

```
def beat_annotations(annotation):
    """ Get rid of non-beat markers """
    """'N' for normal beats. Similarly we can give the input 'L' for left bundle branch block be
        beats. 'A' for Atrial premature contraction. 'V' for ventricular premature contraction.
        escape beat."""

    good = ['N']
    ids = np.in1d(annotation.symbol, good)

    # We want to know only the positions
    beats = annotation.sample[ids]

    return beats

```

Figure 31 Implementation 2

The beat annotations() function will return the indexes of all beats inside a given group (In the code below, the indices of Normal beats have been found).

```
29    def segmentation(records):
30        Normal = []
31        for e in records:
32            signals, fields = wfdb.rdsamp(e, channels = [0])
33
34            ann = wfdb.rdann(e, 'atr')
35            good = ['N']
36            ids = np.in1d(ann.symbol, good)
37            imp_beats = ann.sample[ids]
38            beats = (ann.sample)
39            for i in imp_beats:
40                beats = list(beats)
41                j = beats.index(i)
42                if(j!=0 and j!=(len(beats)-1)):
43                    x = beats[j-1]
44                    y = beats[j+1]
45                    diff1 = abs(x - beats[j])//2
46                    diff2 = abs(y - beats[j])//2
47                    Normal.append(signals[beats[j] - diff1: beats[j] + diff2, 0])
48        return Normal
```

Figure 32 Implementation 3

The beats of each group are segmented using the segmentation() function.

```
1   data = np.array(csv_data)
2   signals = []
3   count = 1
4   peaks =  biosppy.signals.ecg.christov_segmenter(signal=data, sampling_rate = 200)[0]
5   for i in (peaks[1:-1]):
6       diff1 = abs(peaks[count - 1] - i)
7       diff2 = abs(peaks[count + 1]- i)
8       x = peaks[count - 1] + diff1//2
9       y = peaks[count + 1] - diff2//2
10      signal = data[x:y]
11      signals.append(signal)
12      count += 1
13  return signals
14
```

Figure 33 Implementation 4

By mapping each ECG beat, It was able to convert ECG signals into ECG images. The R-peaks in ECG signals were first observed using Python's Biosppy module. To segment a beat after finding the R-peaks, It took the current R-peak and the last R-peak, divided the interval between them by half, and used those signals in the current beat. It did the same thing with the next beat.

```
for count, i in enumerate(array):
  fig = plt.figure(frameon=False)
  plt.plot(i)
  plt.xticks([]), plt.yticks([])
  for spine in plt.gca().spines.values():
     spine.set_visible(False)

  filename = directory + '/' + str(count)+'.png'
  fig.savefig(filename)
  im_gray = cv2.imread(filename, cv2.IMREAD_GRAYSCALE)
  im_gray = cv2.resize(im_gray, (128, 128), interpolation = cv2.INTER_LANCZOS4)
  cv2.imwrite(filename, im_gray)
```

Figure 34 Implementation 5

Here Matplotlib and OpenCV are used to transform these segmented signals into images. Since colors aren't important in ECG signals, It converted them to grayscale images.

```
def cropping(image, filename):

    #Left Top Crop
    crop = image[:96, :96]
    crop = cv2.resize(crop, (128, 128))
    cv2.imwrite(filename[:-4] + 'leftTop' + '.png', crop)

    #Center Top Crop
    crop = image[:96, 16:112]
    crop = cv2.resize(crop, (128, 128))
    cv2.imwrite(filename[:-4] + 'centerTop' + '.png', crop)

    #Right Top Crop
    crop = image[:96, 32:]
    crop = cv2.resize(crop, (128, 128))
    cv2.imwrite(filename[:-4] + 'rightTop' + '.png', crop)

    #Left Center Crop
    crop = image[16:112, :96]
    crop = cv2.resize(crop, (128, 128))
    cv2.imwrite(filename[:-4] + 'leftCenter' + '.png', crop)

```

Figure 35 Implementation 6

```
    #Center Center Crop
    crop = image[16:112, 16:112]
    crop = cv2.resize(crop, (128, 128))
    cv2.imwrite(filename[:-4] + 'centerCenter' + '.png', crop)

    #Right Center Crop
    crop = image[16:112, 32:]
    crop = cv2.resize(crop, (128, 128))
    cv2.imwrite(filename[:-4] + 'rightCenter' + '.png', crop)

    #Left Bottom Crop
    crop = image[32:, :96]
    crop = cv2.resize(crop, (128, 128))
    cv2.imwrite(filename[:-4] + 'leftBottom' + '.png', crop)

    #Center Bottom Crop
    crop = image[32:, 16:112]
    crop = cv2.resize(crop, (128, 128))
    cv2.imwrite(filename[:-4] + 'centerBottom' + '.png', crop)

    #Right Bottom Crop
    crop = image[32:, 32:]
    crop = cv2.resize(crop, (128, 128))
    cv2.imwrite(filename[:-4] + 'rightBottom' + '.png', crop)
```

Figure 36 Implementation 7

Here nine separate cropping methods to six ECG arrhythmia beats (PVC, PAB, RBB, LBB, APC, VEB): left top, middle top, right top, center left, center, center right, left bottom, center bottom, and right bottom have been added. Each cropping method yields two of three ECG image sizes: 96 x 96, 96 x 96, and 96 x 96. The augmented images are then resized to their original resolution of 128 x 128 pixels.

3.7.2 Hardware Implementation

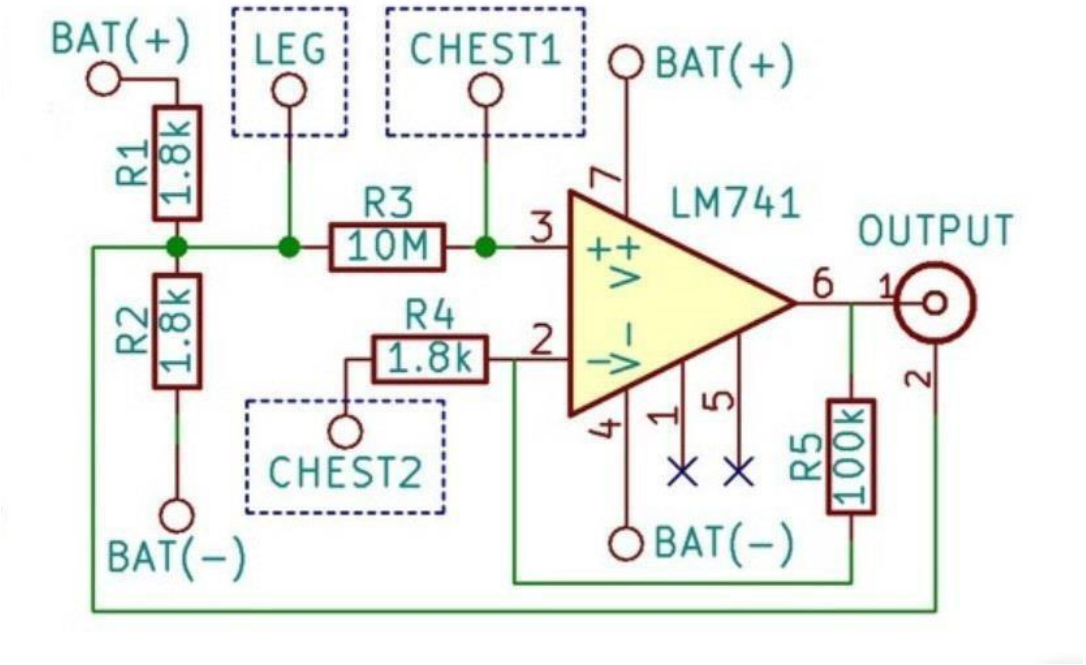

Figure 37 Hardware circuit diagram

From the above architecture, the ECG measuring device was created which measures our ECG and shows in our pc/Laptops. The output is connected to your computer's microphone port. The op-amp used in this hardware isn't anything unusual (LM741). An LM324 (or some other general-purpose op-amp) in a single unit should be sufficient. We have to make sure R1 and R2 are the same value, and R3 is at least 10 MOhm where ‘R’ means resistor.

9V battery has been used to give power to this hardware. It is found that the most comfortable electrodes were those designed for muscle simulators. My friend showed me some muscle stimulator pads that he purchased for a back-pain relief kit that he uses. It instantly figured those pads would be ideal for creating an ECG when it was first seen. They're a little pricey, but they're very convenient, reusable, long-lasting, and deliver fantastic results.

Required Components

- Single Op-amp (LM741)
- Five resistors

- Battery (9V)
- Breadboard
- Wires

3.7.3 Webpage Implementation

```
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>ECG result</title>
    <link href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css" rel="stylesheet">
    <script src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"></script>
    <script src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></script>
    <script src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
    <link href="{{ url_for('static', filename='css/main.css') }}" rel="stylesheet">
</head>

<body>
    <nav class="navbar navbar-dark bg-dark">
        <div class="container">
            <a class="navbar-brand" href="#">Stay Healthy, Stay Happy by- Sandesh Chhetri</a>
            <button class="btn btn-outline-secondary my-2 my-sm-0" type="submit">Help</button>
        </div>
    </nav>
    <div class="container">
        <div id="content" style="margin-top:2em">{% block content %}{% endblock %}</div>
    </div>
</body>

<footer>
    <script src="{{ url_for('static', filename='js/main.js') }}" type="text/javascript"></script>
</footer>

</html>

```

Figure 38 Implementation 8

```
{% extends "base.html" %} {% block content %}

<h2>ECG Arrhythmia Classifier</h2>

<div>
    <form id="upload-file" method="post" enctype="multipart/form-data">
        <label for="imageUpload" class="upload-label">
            Choose...
        </label>
        <input type="file" name="file" id="imageUpload" accept=".csv">
    </form>

    <div class="image-section" style="display:none;">
        <div>
            <div>
            </div>
        </div>
        <div>
            <button type="button" class="btn btn-primary btn-lg " id="btn-predict">Predict!</button>
        </div>
    </div>

    <div class="loader" style="display:none;"></div>

    <h3 id="result">
        <span> </span>
    </h3>

</div>

{% endblock %}

```

Figure 39 Implementation 9

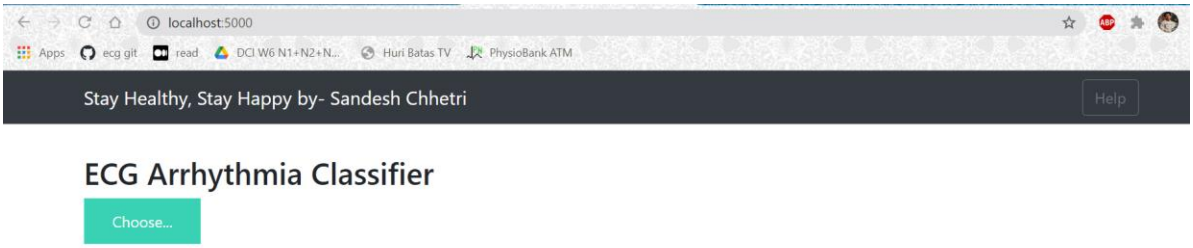

Figure 40 Implementation 10

Chapter 4: Testing and Analysis

The development must go through a testing and analysis process after the planning phase is completed. This is one of the most critical and essential phases of project development. This process is completed with all project construction before it is released to the general public in order to determine if all of the project's functionality and features are functioning properly. After the testing is completed, an analysis can be performed to better identify flaws and fixes can be made for the final process.

3.8 Test Plan

3.8.1 Unit Testing, Test Plan

Unit testing is a technique for evaluating the smallest piece of code, referred to as a unit. The key aim is to ensure that each software unit works as expected. The most critical aspect of Unit Testing is to concentrate on the tests that impact the system's actions rather than creating test cases for all (INAN, July, 2020).

Test plans for unit testing are mentioned below in the table.

Test Cases	Objectives
1	To test whether all the connections are properly connected.
2	To test whether the ECG software reads the input signal.
3	To test if we can capture our ECG signal.
4	To test if we can upload our ECG data.
5	To test if PREDICT button processes our ECG data.
6	To test if our ECG signal can detect PVC etc.

Table 2 Unit Testing, Test plan

3.8.2 System Testing, Test Plan

System testing is a web application testing process that focuses on testing the whole system and recording how it responds. This process evaluates the system's compliance by testing the finished project.

Test Cases	Objectives
1	To read ECG data from device and display in software.
2	To upload sample ECG data and show results.

Table 3 System Testing, Test Plan

3.9 Unit Testing

3.9.1 Test Case 1

Objective	To test whether all the connections are properly connected.
Expected Result	Properly designed hardware.
Actual Result	Photo of properly designed hardware.
Conclusion	Successful

Table 4 Unit Test case 1

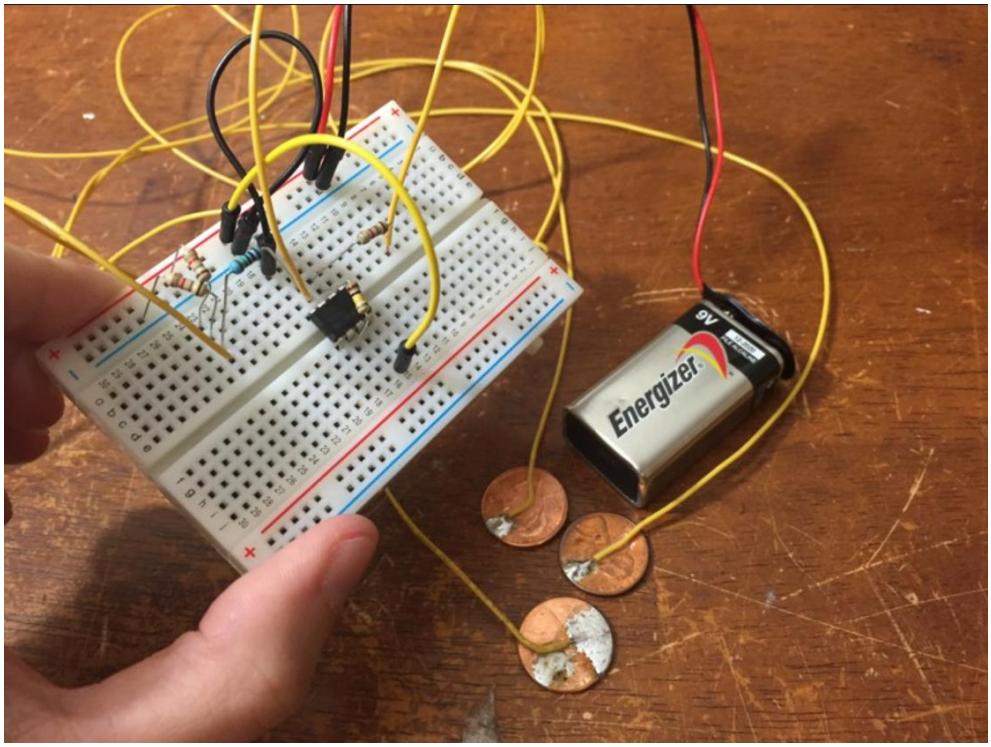

Figure 41 Hardware Design Test 1

3.9.2 Test Case 2

Objective	To test whether the ECG software reads the input signal.
Expected Result	Recording live ECG signal
Actual Result	Live ECG record
Conclusion	Successful

Table 5 Unit test case 2

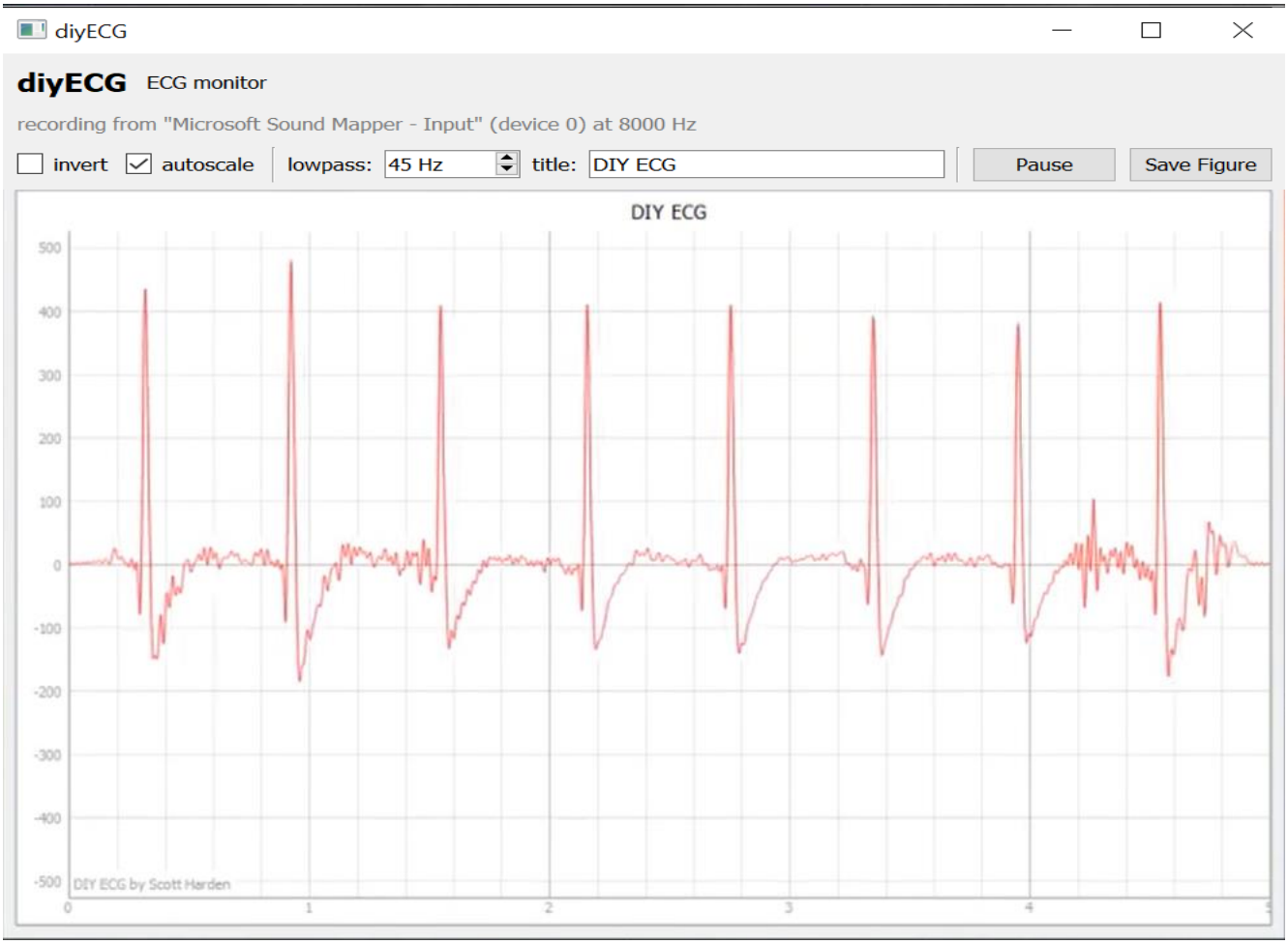

Figure 42 Live ECG data

3.9.3 Test Case 3

Objective	To test if we can capture our ECG signal
Expected Result	Save a jpg of our ECG
Actual Result	Jpg Images is seen
Conclusion	Successful

Table 6 Unit Test case 3

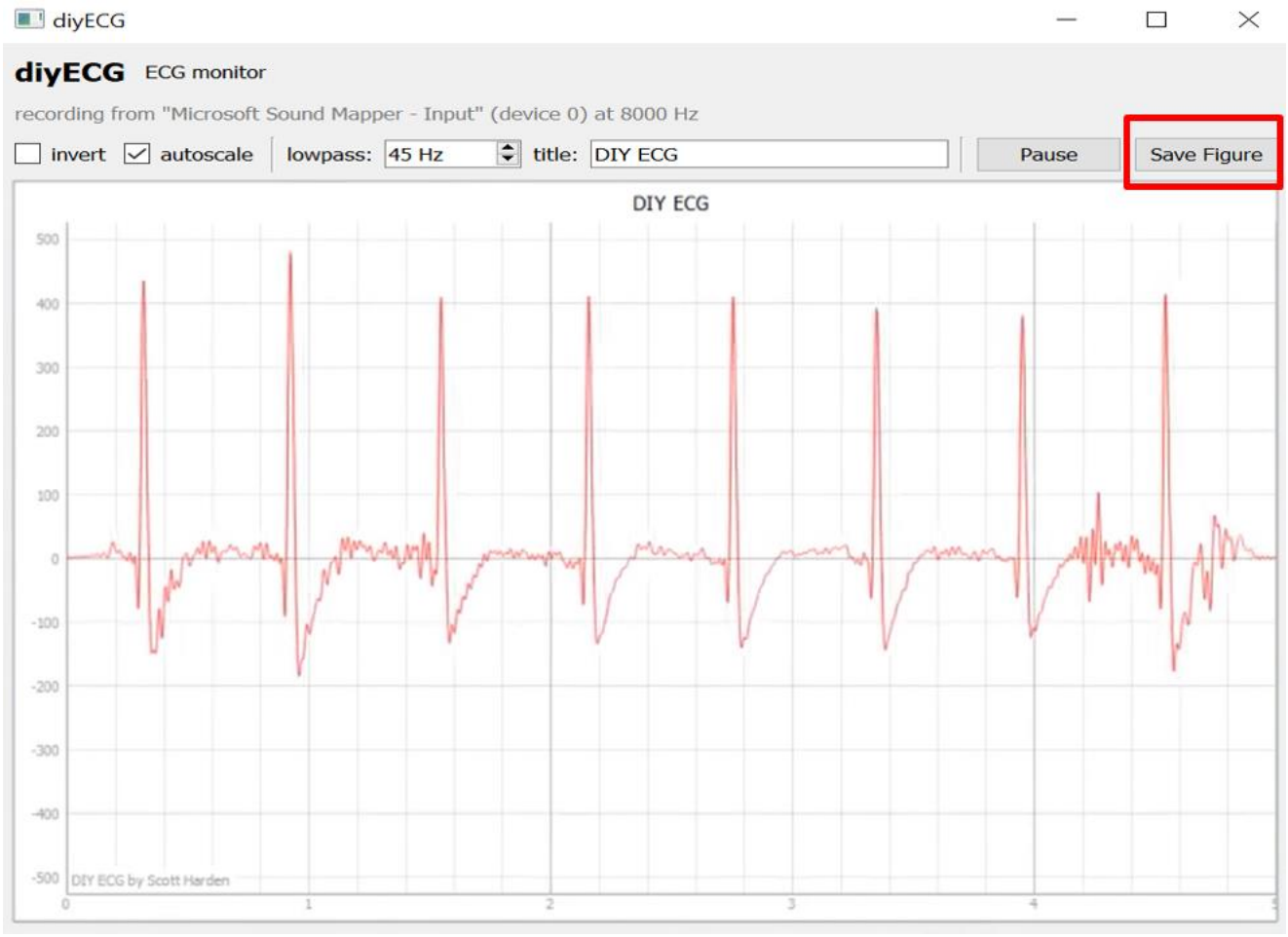

Figure 43 Saving ECG picture

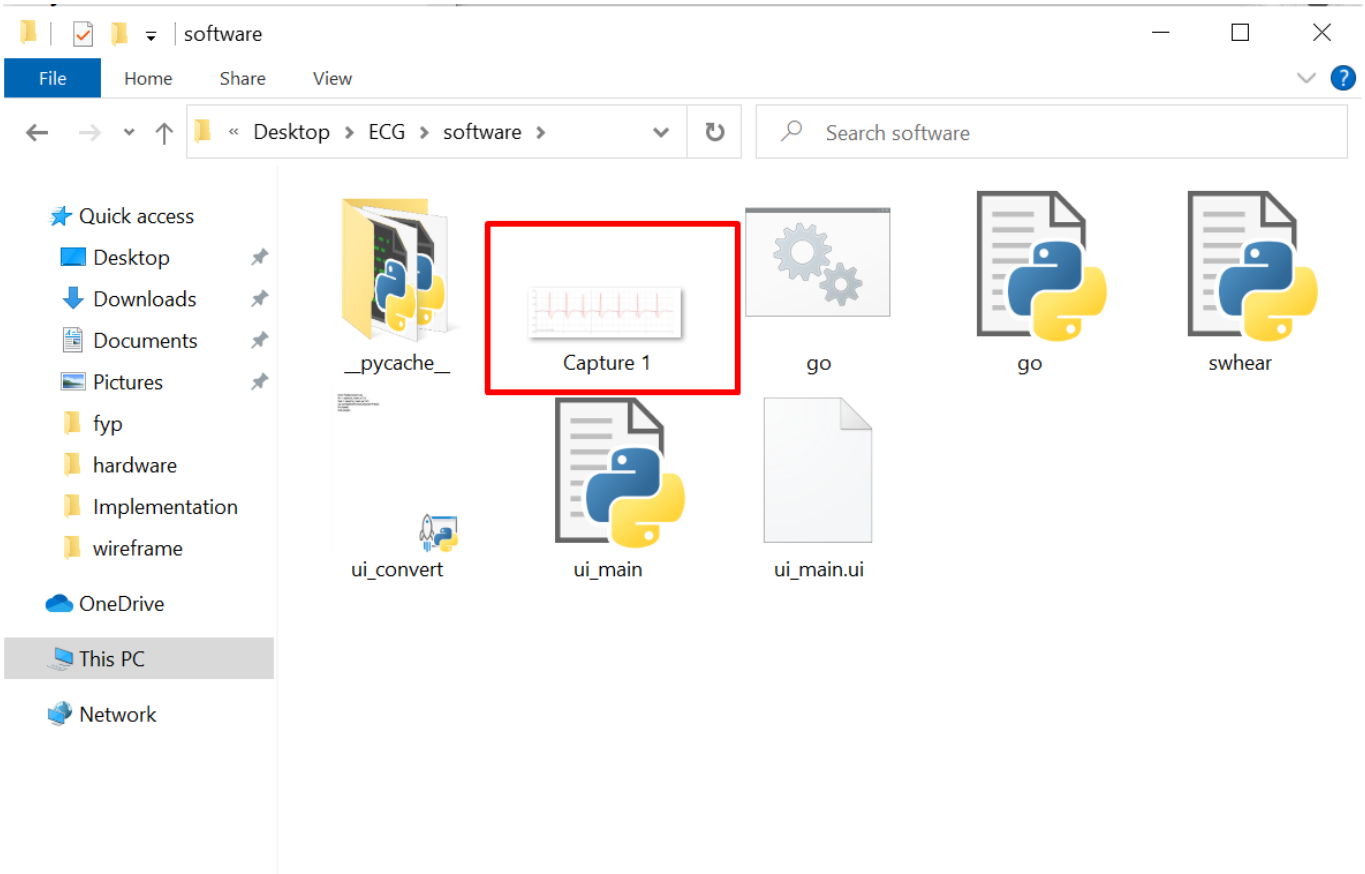

Figure 44 ECG image saved

3.9.4 Test case 4

Objective	To test if we can upload our ECG data
Expected Result	Upload ECG data in webpage
Actual Result	ECG data uploaded
Conclusion	Successful

Table 7 Unit test case 4

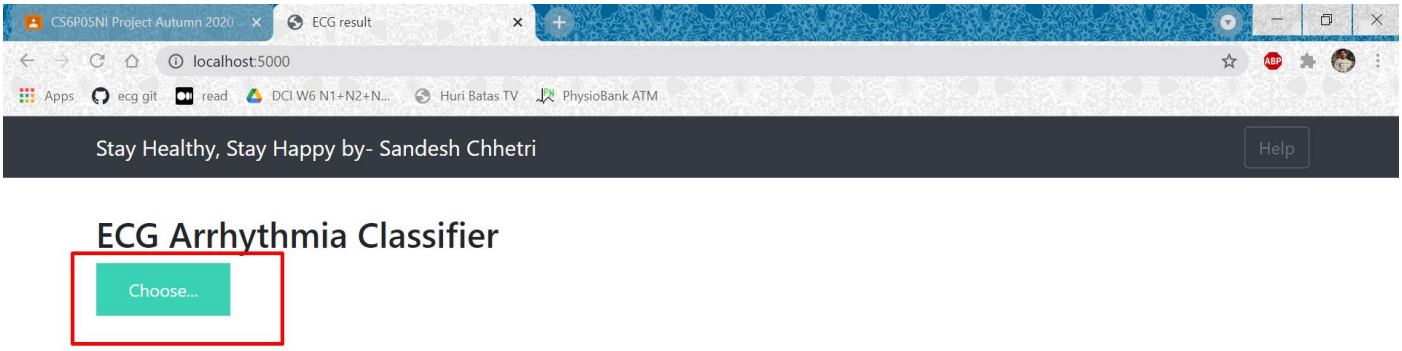

Figure 45 Uploading ECG data

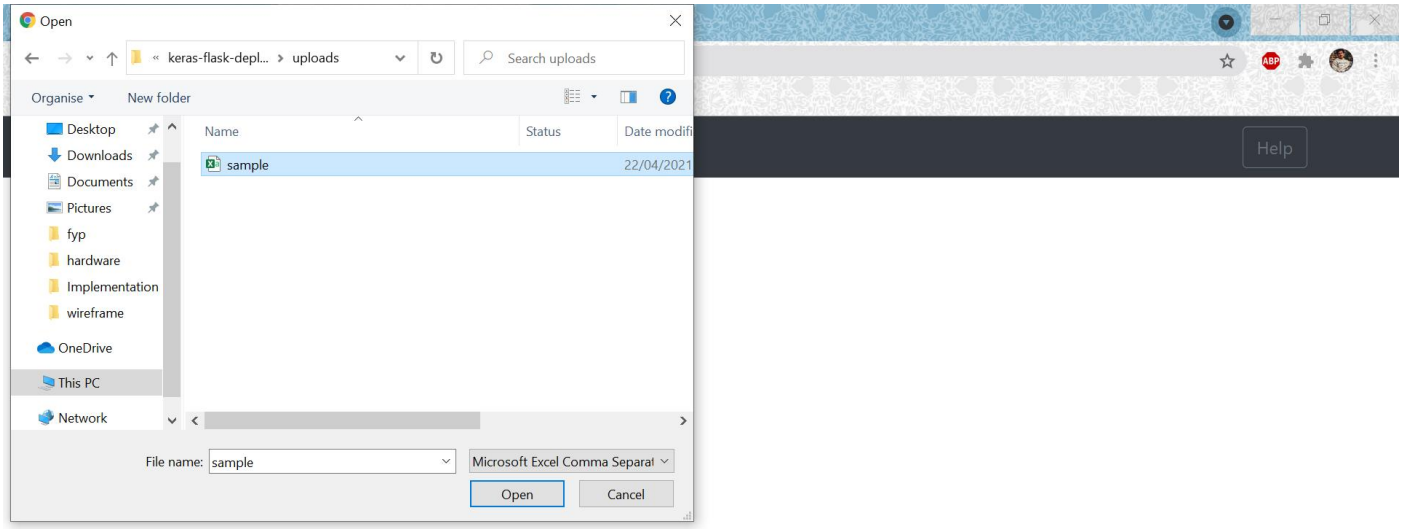

Figure 46 Selecting sample ECG data

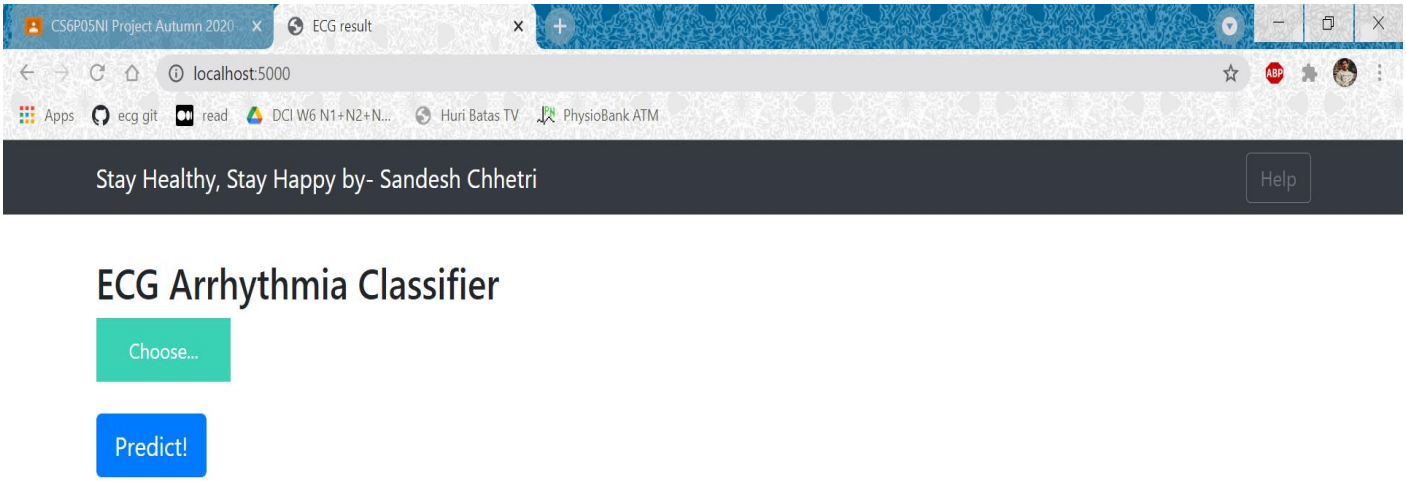

Figure 47 ECG data uploaded 1

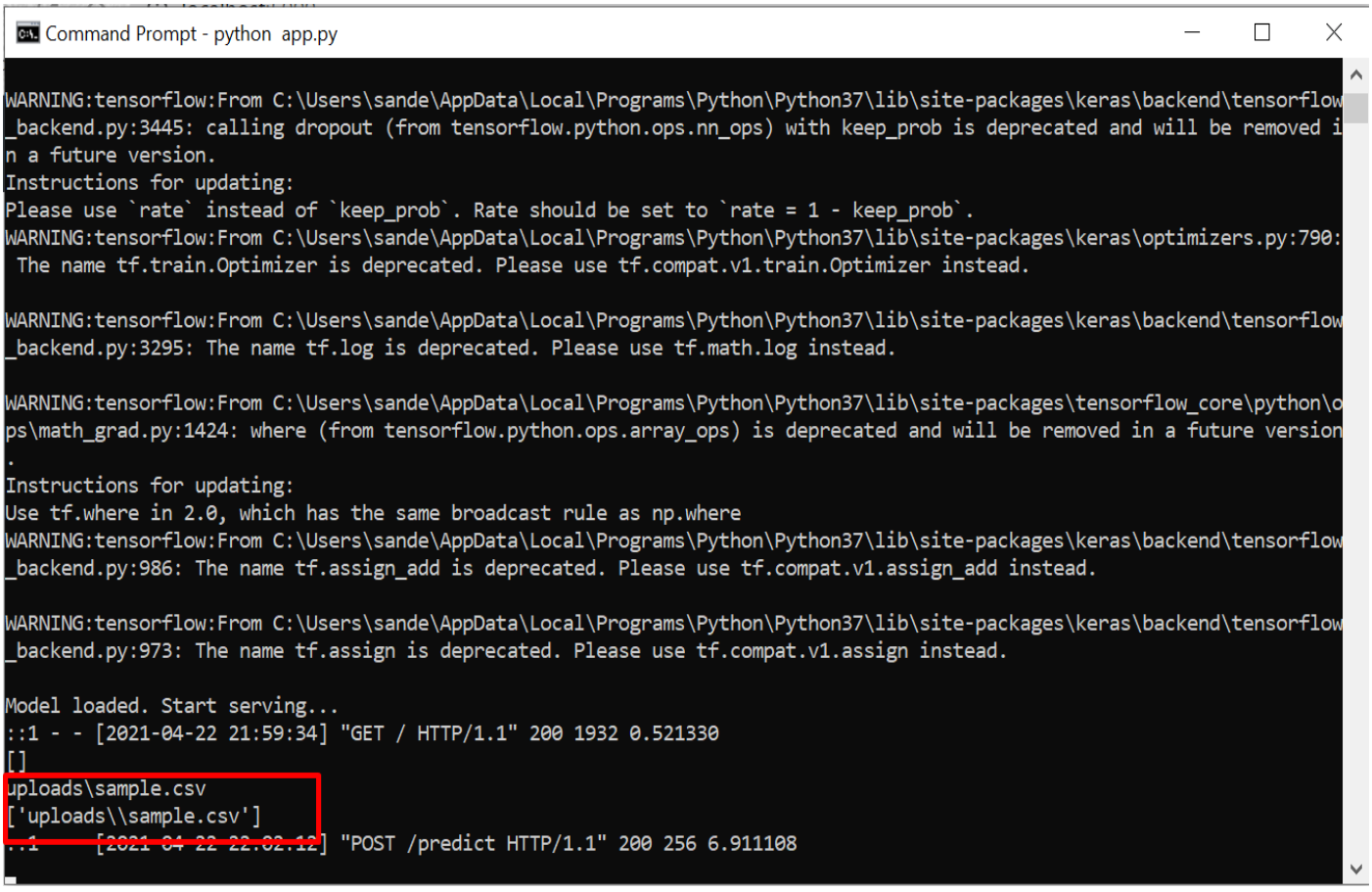

Figure 48 ECG data uploaded 2

3.9.5 Test case 5

Objective	To test if PREDICT button processes our ECG data
Expected Result	Predict ECG data
Actual Result	When clicking on predict button, it is processing our ECG data.
Conclusion	Successful

Table 8 Unit test case 5

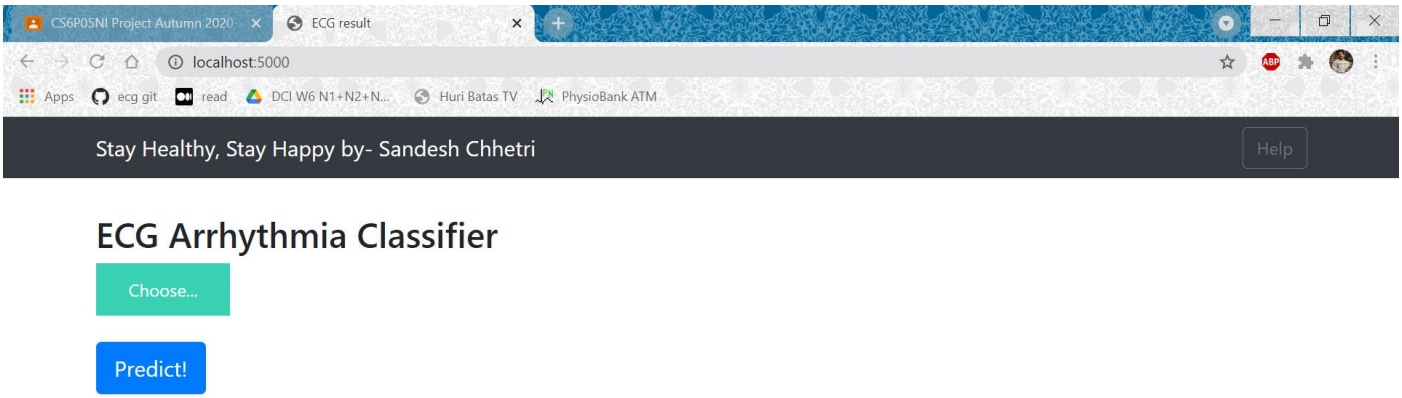

Figure 49 Clicking on predict button

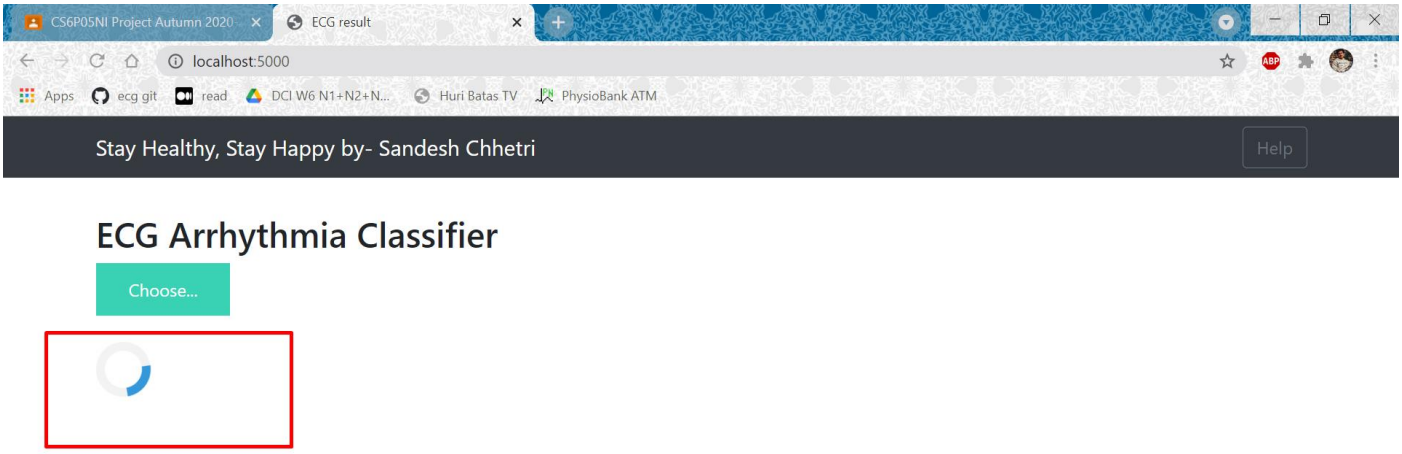

Figure 50 ECG data is being processed

3.9.6 Test case 6

Objective	To test if our ECG signal can detect PVC
Expected Result	Show result about PVC
Actual Result	Result of PVC is seen
Conclusion	Successful

Table 9 Unit test case 6

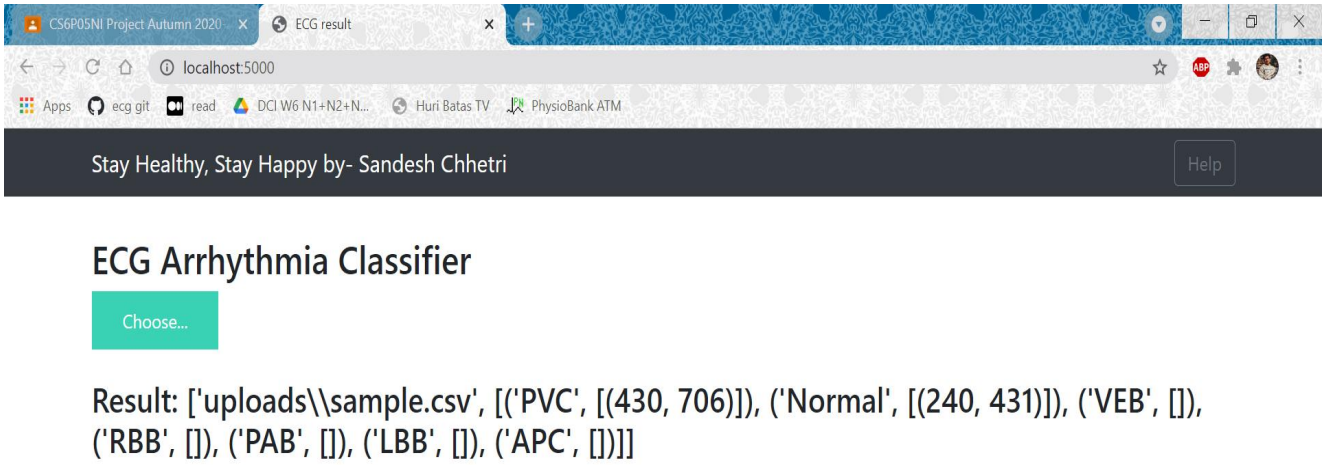

Figure 51 PVC is seen

3.10 System Testing

3.10.1 Test case 1

Objective	To read ECG data from device and display in software
Expected Result	Show live ECG in software
Actual Result	Live ECG signal is seen
Conclusion	Successful

Table 10 System test case 1

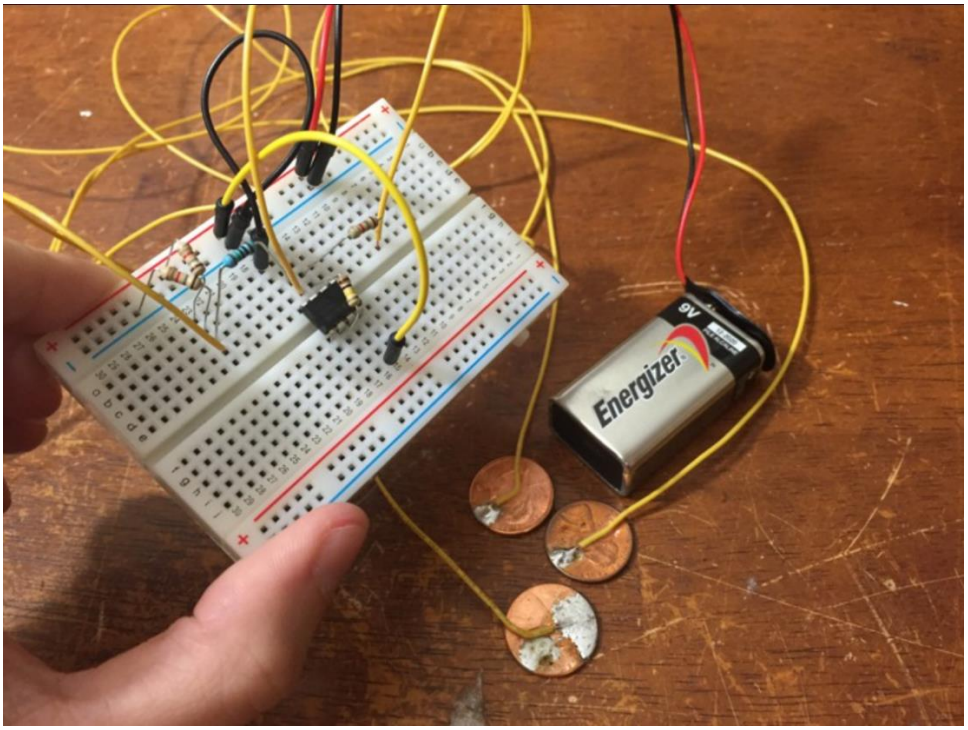

Figure 52 Testing connection

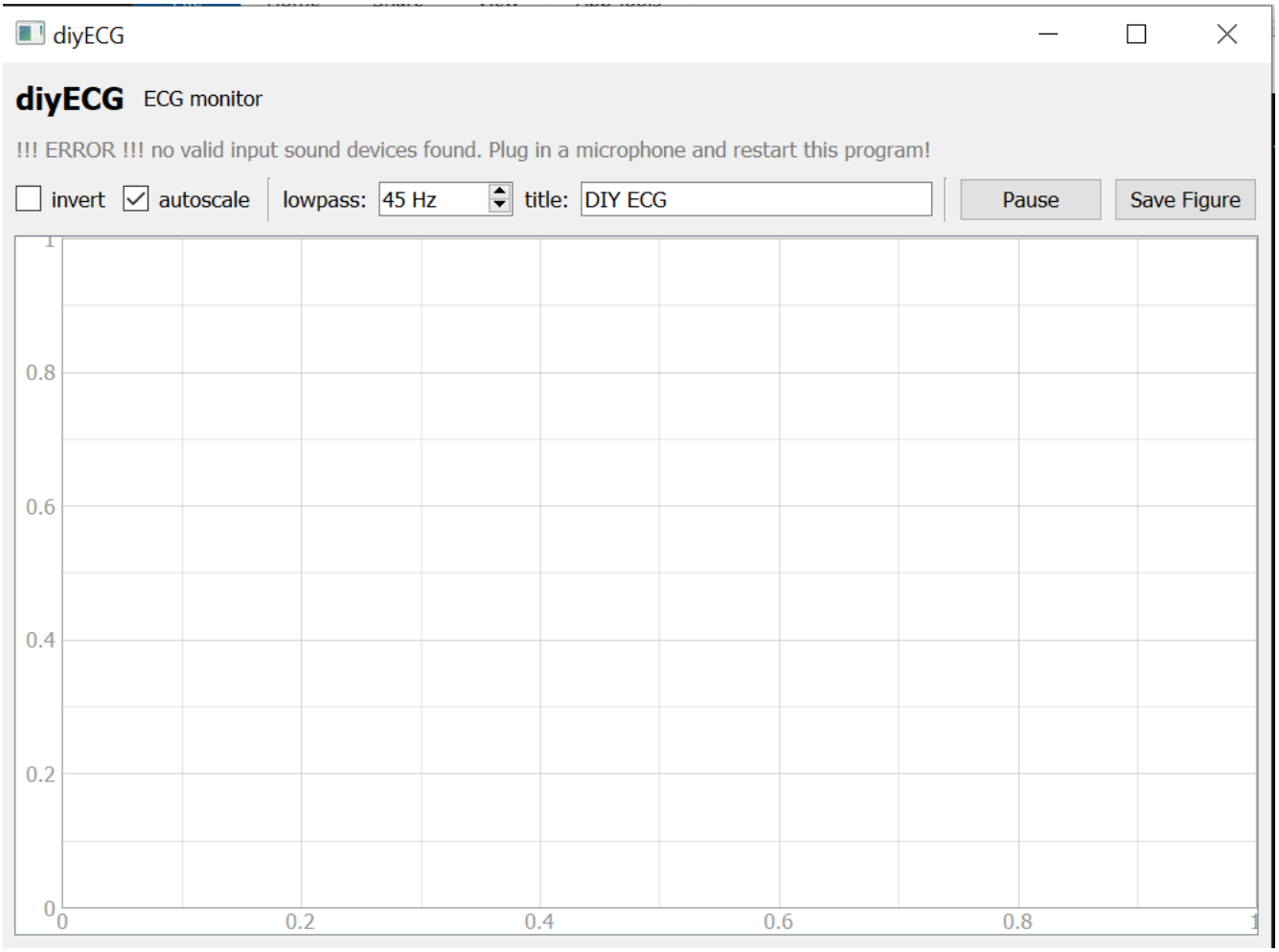

Figure 53 Before plugin hardware

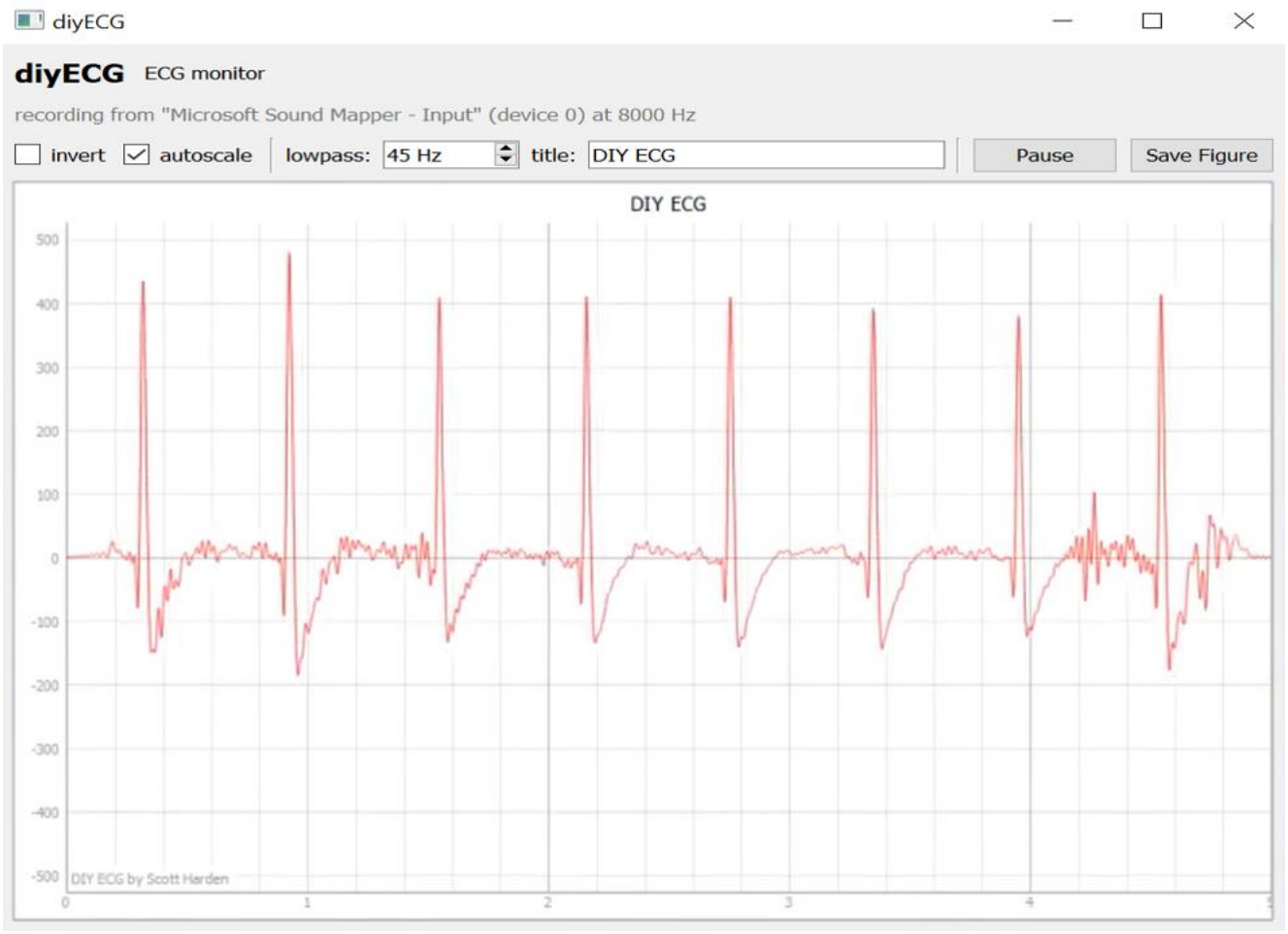

Figure 54 After plugin hardware

3.10.2 Test Case 2

Objective	To upload sample ECG data and show results
Expected Result	Upload ECG data and show the result of data
Actual Result	Data is uploaded and PVC result is seen
Conclusion	Successful

Figure 55 System test case 2

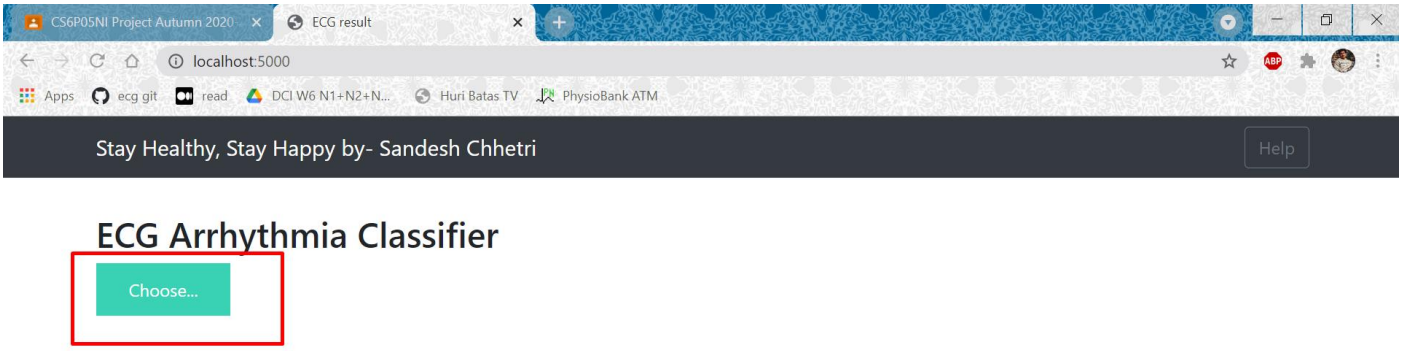

Figure 56 Uploading a ECG data in webpage

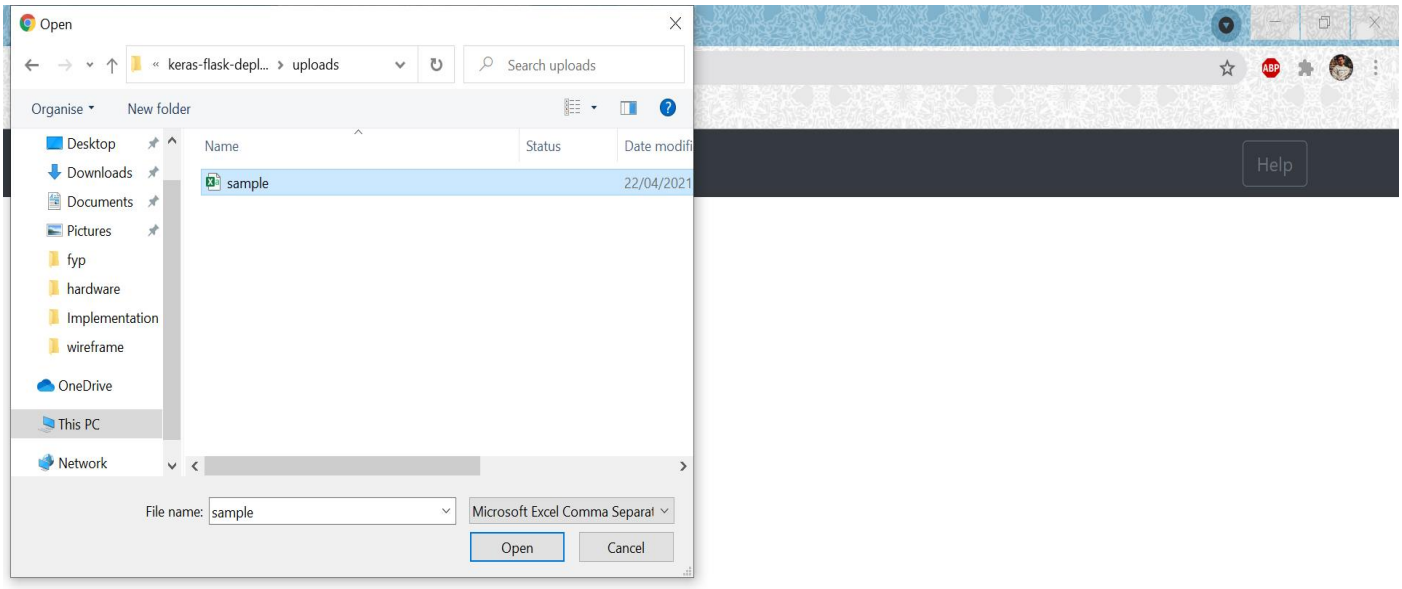

Figure 57 Uploading sample.csv data in webpage

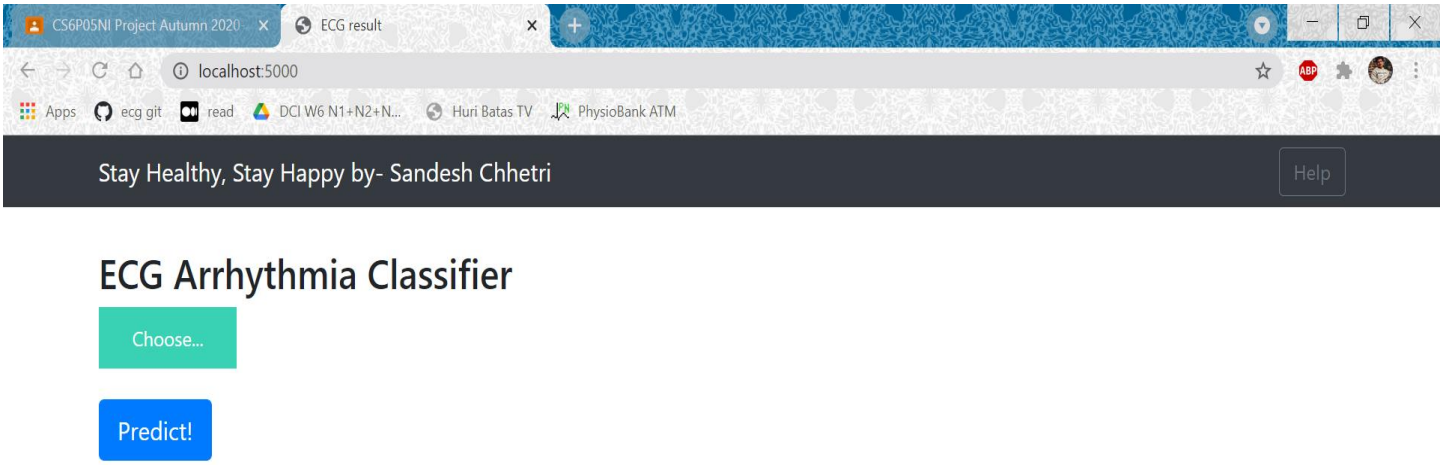

Figure 58 ECG data has been uploaded in webpage

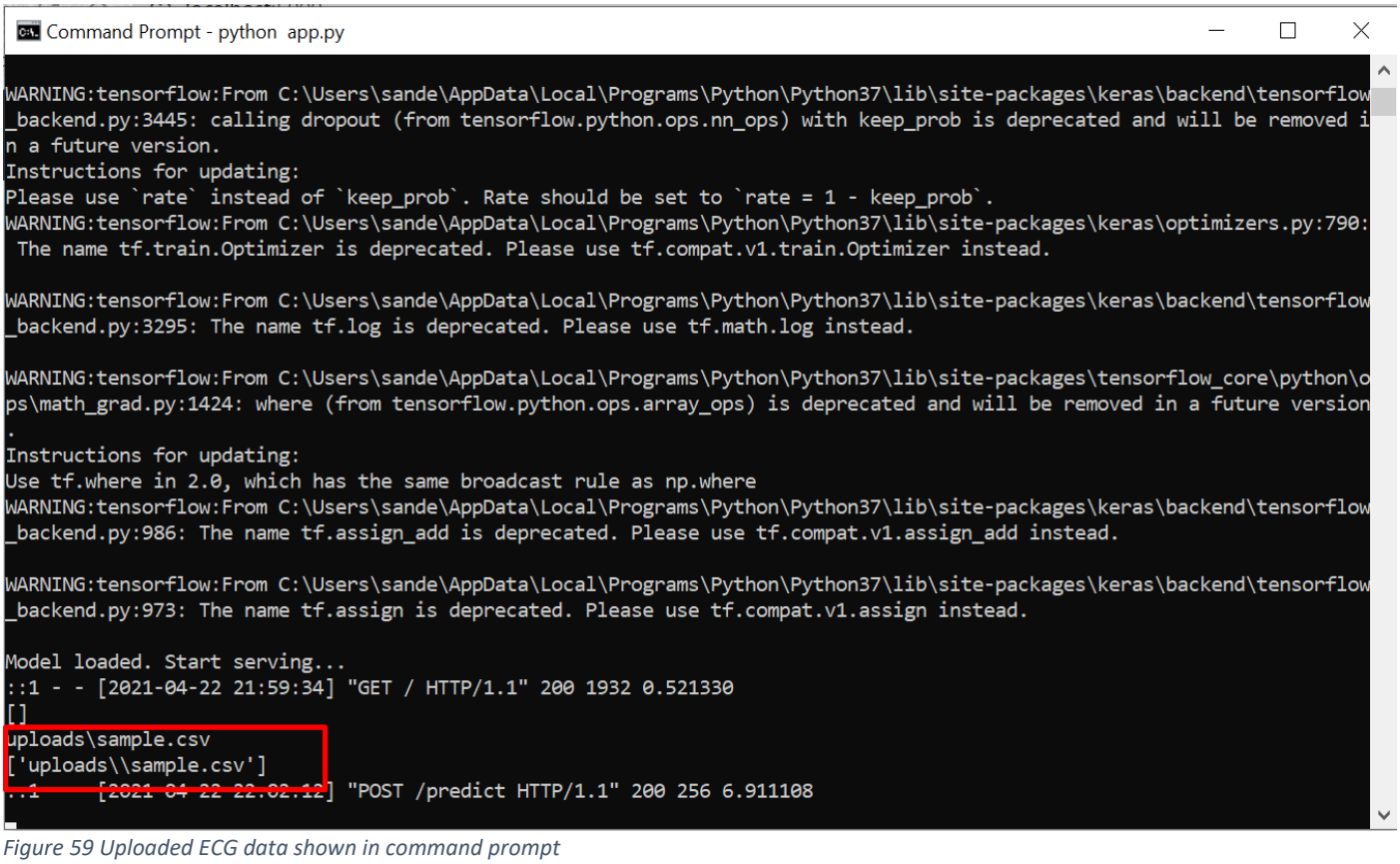

Figure 59 Uploaded ECG data shown in command prompt

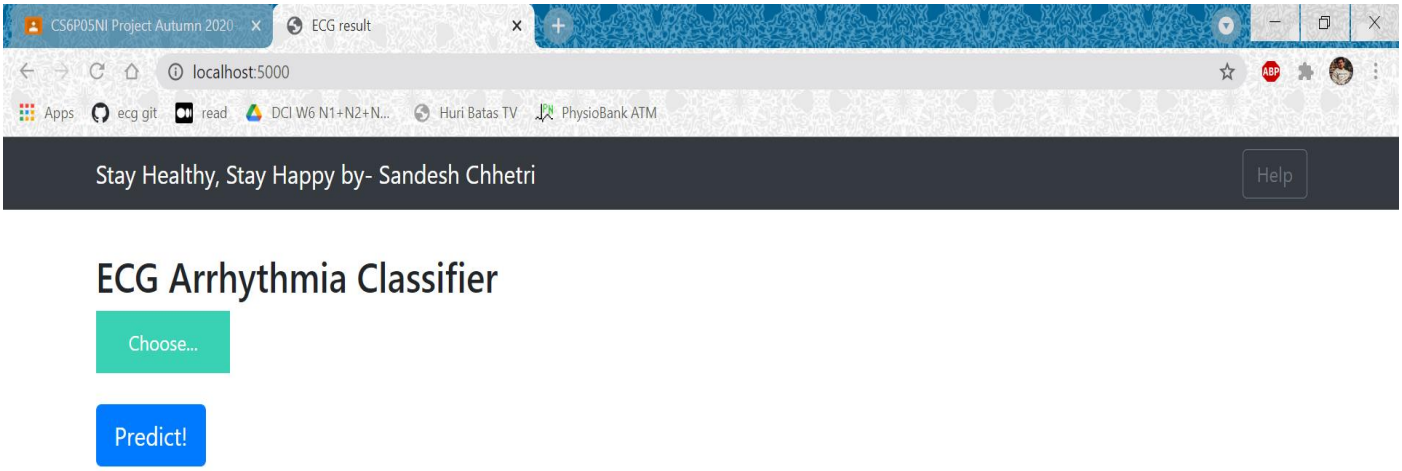

Figure 60 Clicking in predict button

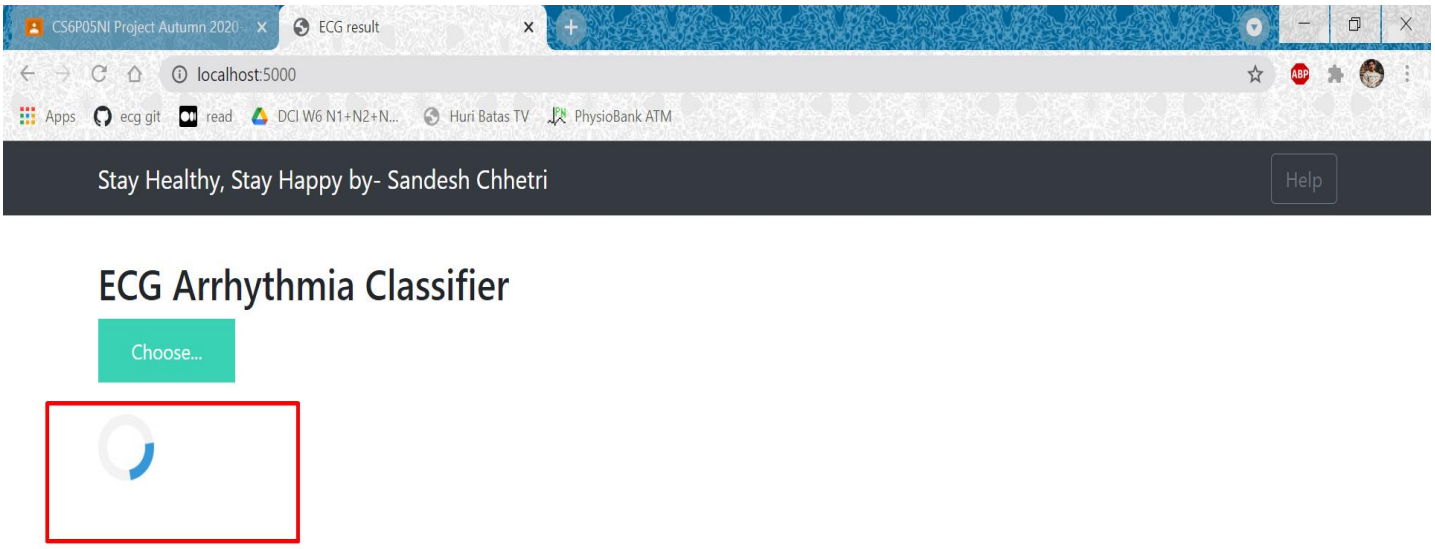

Figure 61 Processing data before result

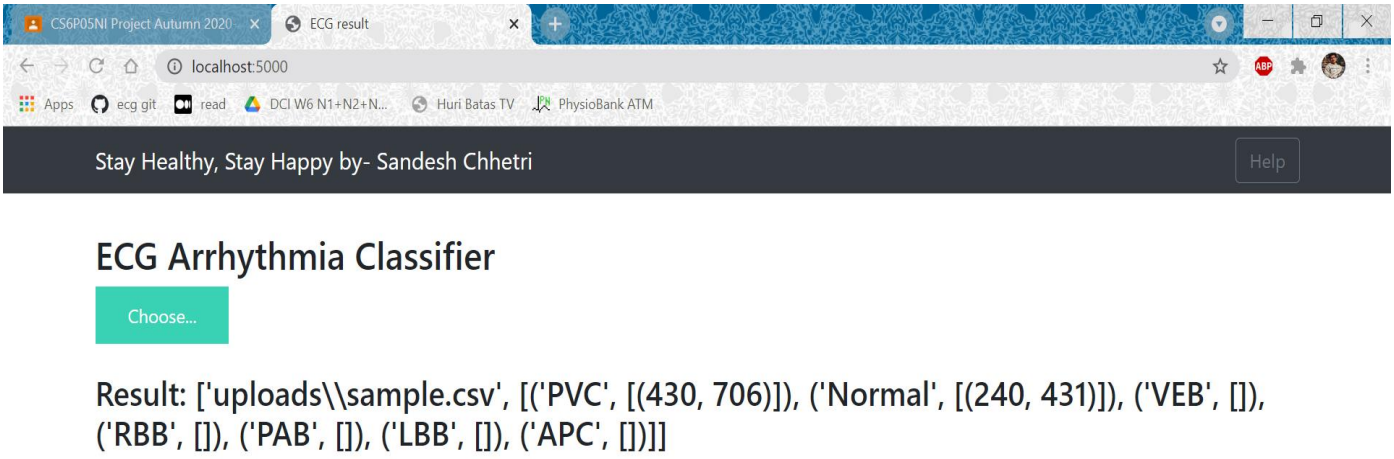

Figure 62 PVC result is seen

3.11 Critical Analysis

After performing unit and device testing on the majority of the project's functions and functionality, it was discovered that all of the experiments were positive. Obtaining outcomes without loss is a significant accomplishment when building a project, since there are several points where the project could collapse. When people see their live ECG, they felt happy and satisfied with the software.

Due to unfortunate situation of the global pandemic ‘COVID-19’, some components of hardware were not available due to which I had to compromise with other equipment as their replacement because of which the ECG signal that we get is not so accurate.

Chapter 5: Conclusion

Finally, ECG monitoring and classification were successfully completed with the combination of all the features identified and extracted after review of pre-survey findings and study into the country's current health situation. ECG monitoring is a project that is helpful to basically all in the general population. It gives the users a great experience.

In terms of the technical aspects, this program was successfully created with Python's TensorFlow software library, and the front end was created using HTML5, Javascript, and CSS. By combining both of these elements, the project was completed.

5.1 Legal, Social and Ethical issues

5.1.1 Legal issues

Legal problems arise as a result of violations of laws, codes, and regulations. Intellectual private property, intellectual property protections such as trademarks, copyrights, and licenses, terms of use, and privacy policy are some of the legal challenges that applications and software's face today. The design, as well as the whole task, does not break any laws that will result in legal action. The platform and tools used during development are both open source and accessible online. There was no use of pirated programs and all of the available services are mentioned in the report. ECG monitoring and its classification did not break any of these types of legal problems when creating this project.

5.1.2 Social issues

Social problems are those that concern individuals or groups of individuals living in a community. The project as a whole would not raise any social problems that might arise in society. Since no category of people is prohibited from using this ECG classification program, it does not reflect or appear to represent any form of inequality between races.

This software can benefit not just to individuals but also the government by raising the country's health consciousness. As a result, this initiative and the whole program have little positive effects on society and are open to everyone.

5.1.3 Ethical issues

Ethical issues are ones that begin with deciding if a certain task is correct or incorrect. Since all materials taken or copied from the internet are accurately referenced, this initiative does not raise any legal concerns. No other people's codes have been used without their permission, and they are not claimed as mine.

The only ethical concern that this project could create is that it could reduce the number of people who visit hospitals for heart checks, and that individuals may become less inclined to visit medical facilities so they can monitor their ECG at home.

5.2 Advantages

ECG Monitoring and Classification come with all these functionality and benefits for the general public.

The below are some of the project's benefits:

- The main feature of this project is that it will be available at a very low cost because its hardware design and web application are built focusing on this. Now, everyone can easily buy it and use it.
- As the device is small and light, we can easily carry it anywhere so it is easily portable.
- We all know that there are not proper resources and methods in rural areas for checking heart beat and its problems. So, this project can also be easily used in rural areas too.
- People can easily examine their heart beats and check their heart condition frequently.
- Another main feature of this project is that it is very user friendly. Anyone with basic knowledge about computers can use it. The process and the webpage are also simple and easy to use.
- This project is less time consuming because immediately when you plugin the jack, ECG signal can be seen. Just by clicking save picture it will save your ECG and after uploading on the website no time result will be in front of us.

5.3 Limitations

Each software program and hardware has advantages and disadvantages. As a result, there are certain drawbacks to this ECG monitoring system and program, which are described below:

- The main drawback of this project is that it can only detect certain number or types of arrhythmia (NOR, PVC, PAB, RBB, LBB, APC, PAB).

- Due to global pandemic, some components were not available due to which we had to compromise on other equipment for the replacement of the hardware needed for project. So, the ECG data is not accurate as the real ECG machine.
- We have focused on minimizing cost so the hardware component use in this project might require frequent repair.

5.4 Future Work

Due to a lack of implementation time and the COVID-19 scenario, the device was unable to include some of the desired features.

Some of the features are:

- Firstly, there are a lot of hardware components that needs to be upgraded for better accurate result.
- Adding more features like recording ECG data for long period of time.
- Managing hardware components properly make it easy for users to use it.
- Add more features for webpage like contact field, login page, proper homepage about ECG data and different menu bar options.
- Try to detect many other arrythmia than listed in this project.

Chapter 6: References

Bernard, J., February, 2018. *Introducing Spyder, the Scientific PYthon Development EnviRonment.* [Online]
Available at: https://www.linuxjournal.com/content/introducing-spyder-scientific-python-development-environment
[Accessed 19 04 2021].

Charvat, J., 2003. *The Project management methodologies: Selecting, implementing and supporting methodologies and processes for projects..* 1st ed. Hoboken, NJ: John Wiley & Sons Inc..

Dalen, j. E., Alpert, J. S. & Goldberg, R. J., 2014. The Epidemic of the 20th Century.. *Coronary Heart Disease,* 127(9), pp. 807-12.

G, H., Z, L., L, V. D. M. & K Q, W., 2017. Densely Connected Convolutional Networks.. *IEEE Conference on Computer Vision and Pattern Recognition,* daoi 10.1109(CVPR.243), pp. 2261-2269.

Hannah, J., April, 2021. *What Exactly Is Wireframing? A Comprehensive Guide.* [Online]
Available at: https://careerfoundry.com/en/blog/ux-design/what-is-a-wireframe-guide/
[Accessed 20 04 2021].

Harden, S. W., August, 2016. *DIY ECG with 1 op-amp.* [Online]
Available at: https://swharden.com/blog/2016-08-08-diy-ecg-with-1-op-amp/
[Accessed 22 03 2021].

Huang, G. a. L. Z. a. V. D. M. L. a. W. K. Q., 2017. Densely Connected Convolutional Networks. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 10.1109(CVPR.2017.243), pp. 2261-2269.

INAN, E., July, 2020. *What is Unit Testing?.* [Online]
Available at: https://medium.com/hardwareandro/what-is-unit-testing-435e8134b16e
[Accessed 20 04 2021].

Letter, H. H., April, 2010. *Exercise stress test.* [Online]
Available at: https://www.health.harvard.edu/heart-health/exercise-stress-test
[Accessed 13 04 2021].

Lindstorm, G., 2005. Programming with Python. *IT Professional,* 7(5), pp. 10-16.

Littlefield, B., February, 2019. *The coolblueweb Guide to Agile Methodology.* [Online]
Available at: https://coolblueweb.com/blog/guide-to-agile-methodology/
[Accessed 28 03 2021].

MBA Skool Team, 2018. *Prototype.* [Online]
Available at: https://www.mbaskool.com/business-concepts/it-and-systems/13712-prototype.html
[Accessed 02 04 2021].

Medina, H. & Farid, N.-A., 2015. An ECG T-wave Anomalies Detection Using a Lightweight Classification Model for Wireless Body Sensors. *IEEE ICC 2015 - Workshop on ICT-enabled services and technologies for eHealth and Ambient Assisted Living,* 10.1109(ICCW.2015.7247191), pp. 278-283.

Moody, G. B. & Mark, R. G., 2001. The impact of the MIT-BIH Arrhythmia Database. *IEEE Engineering in Medicine and Biology Magazine,* 20(3), pp. 45-50.

Patel, A. M. et al., 2000. Treatment of underlying atrial fibrillation: paced rhythm obscures recognition. *Journal of the American College of Cardiology,* 36(3), pp. 784-787.

Sami, M., March, 2012. *Software Development Life Cycle Models and Methodologies.* [Online]
Available at: https://melsatar.blog/2012/03/15/software-development-life-cycle-models-and-methodologies/
[Accessed 02 04 2021].

Selner, M., September, 2018. *Atrial Premature Complexes.* [Online]
Available at: https://www.healthline.com/health/atrial-premature-complexes
[Accessed 19 04 2021].

Sharma, L., April 17, 2016. *WaterFall Model.* [Online]
Available at: https://www.toolsqa.com/software-testing/waterfall-model/
[Accessed 09 04 2021].

Singh, A., July, 2018. *ECG arrhythmia classification using a 2-D.* [Online]
Available at: https://medium.datadriveninvestor.com/ecg-arrhythmia-classification-using-a-2-d-convolutional-neural-network-33aa586bad67
[Accessed 16 04 2021].

TechTarget Contributor, 2008. *DEFINITION FLOWCHART.* [Online]
Available at: https://whatis.techtarget.com/definition/flowchart
[Accessed 21 04 2021].

Chapter 7: Appendix

7.1 Appendix A: Pre-Survey

7.1.1 Pre-Survey form

Figure 63 Pre-Survey form 1

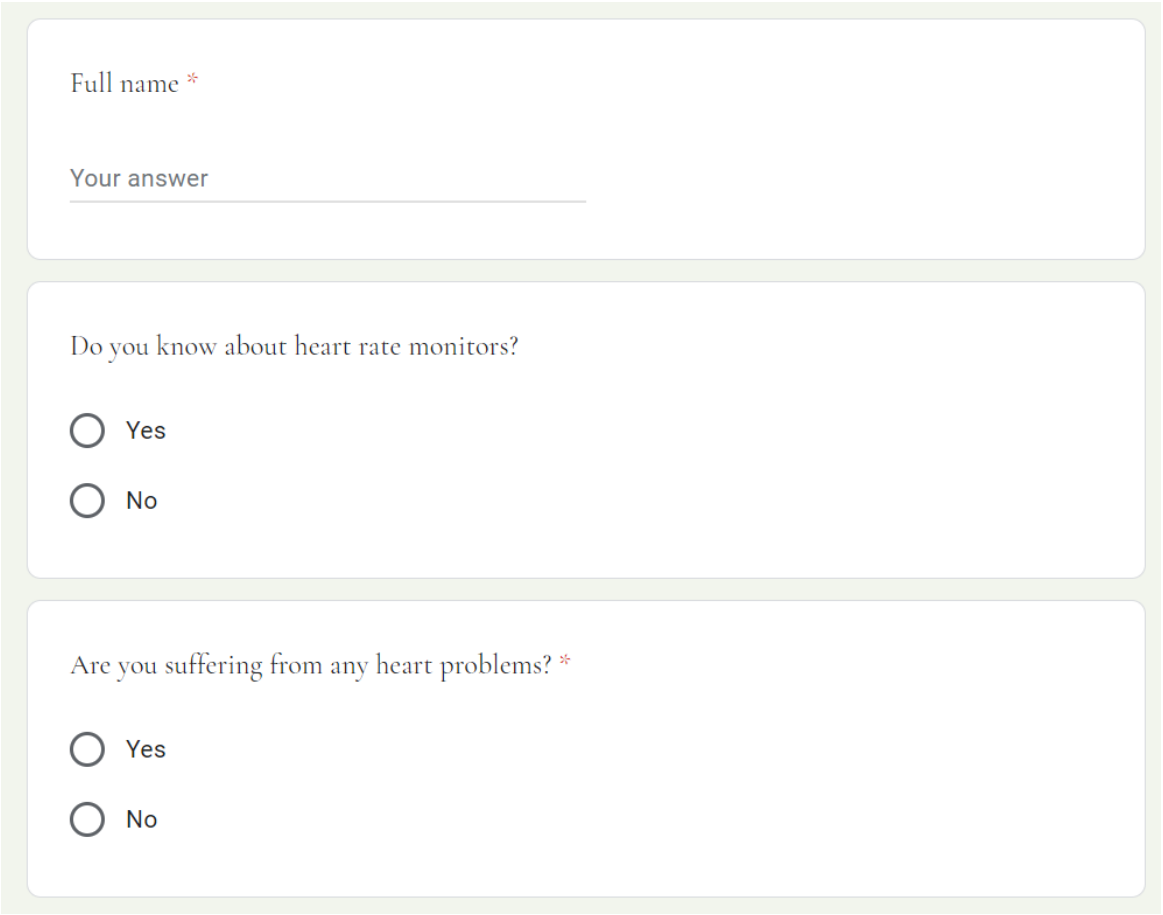

Figure 64 Pre-Survey form 2

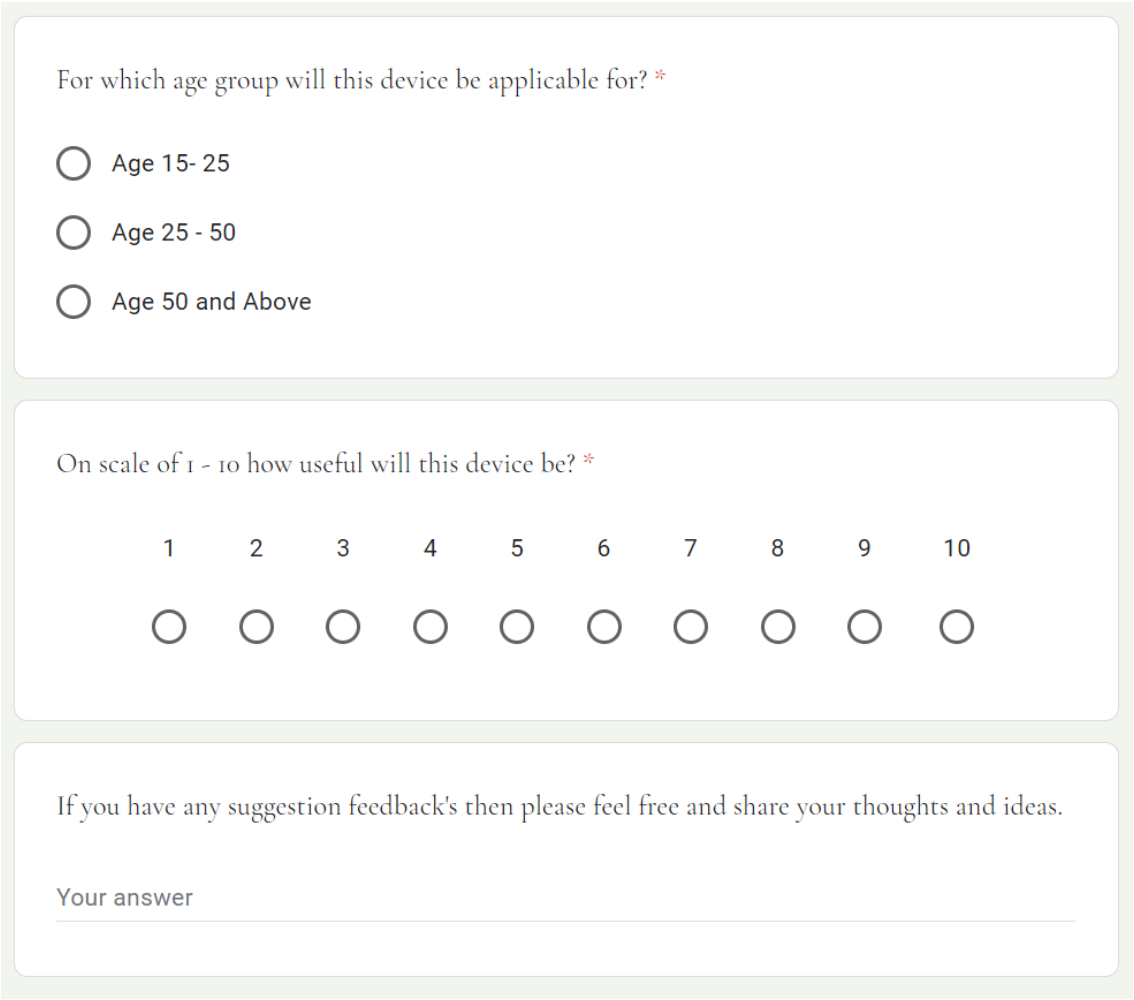

Figure 65 Pre-Survey form 3

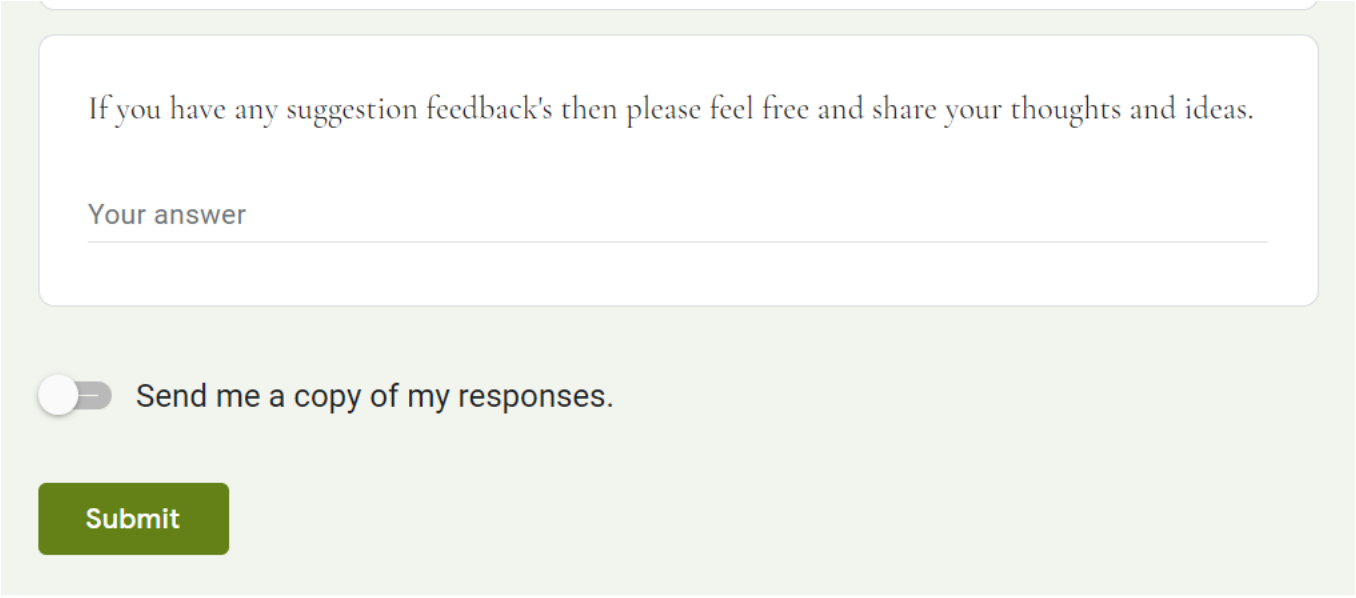

Figure 66 Pre-Survey form 4

7.1.2 Sample of filed Pre-Survey forms

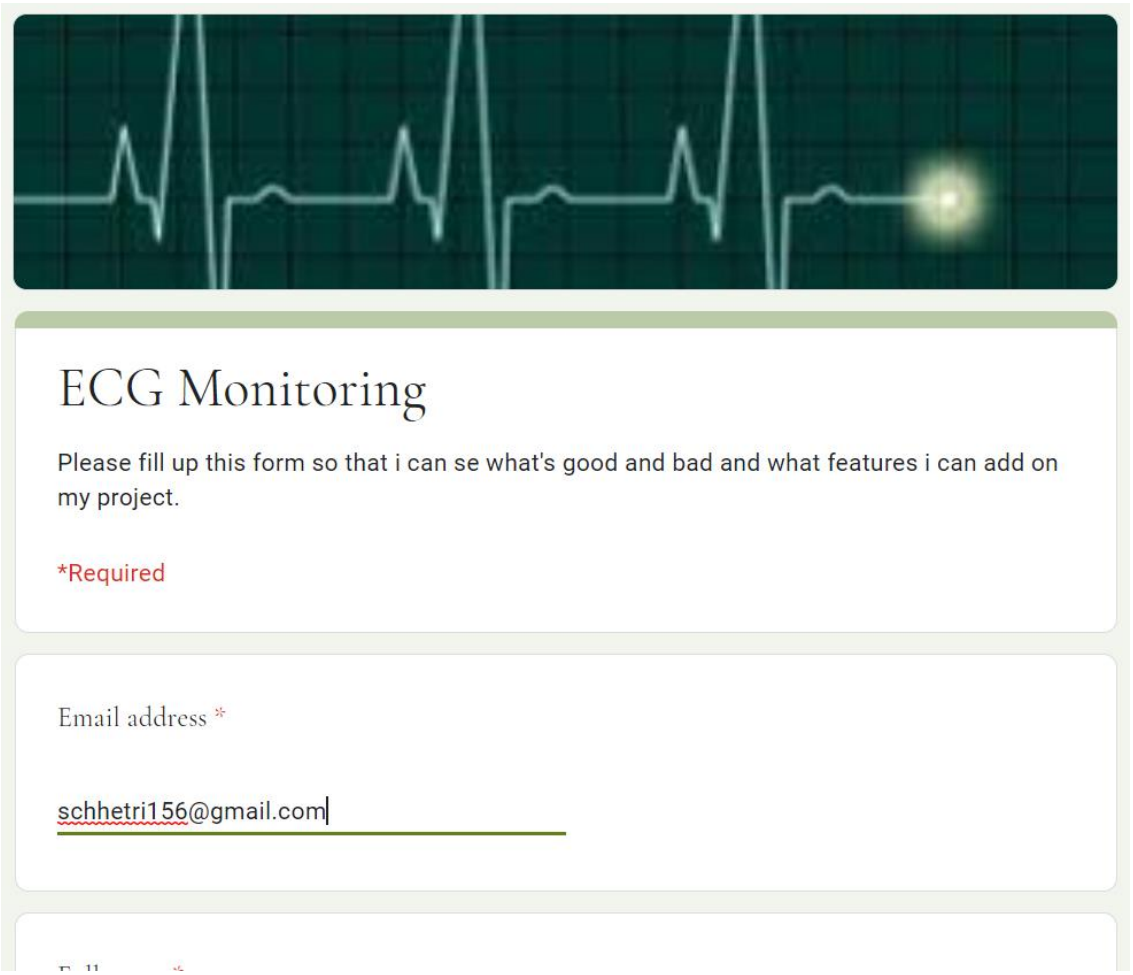

Figure 67 Filled Pre-Survey form 1

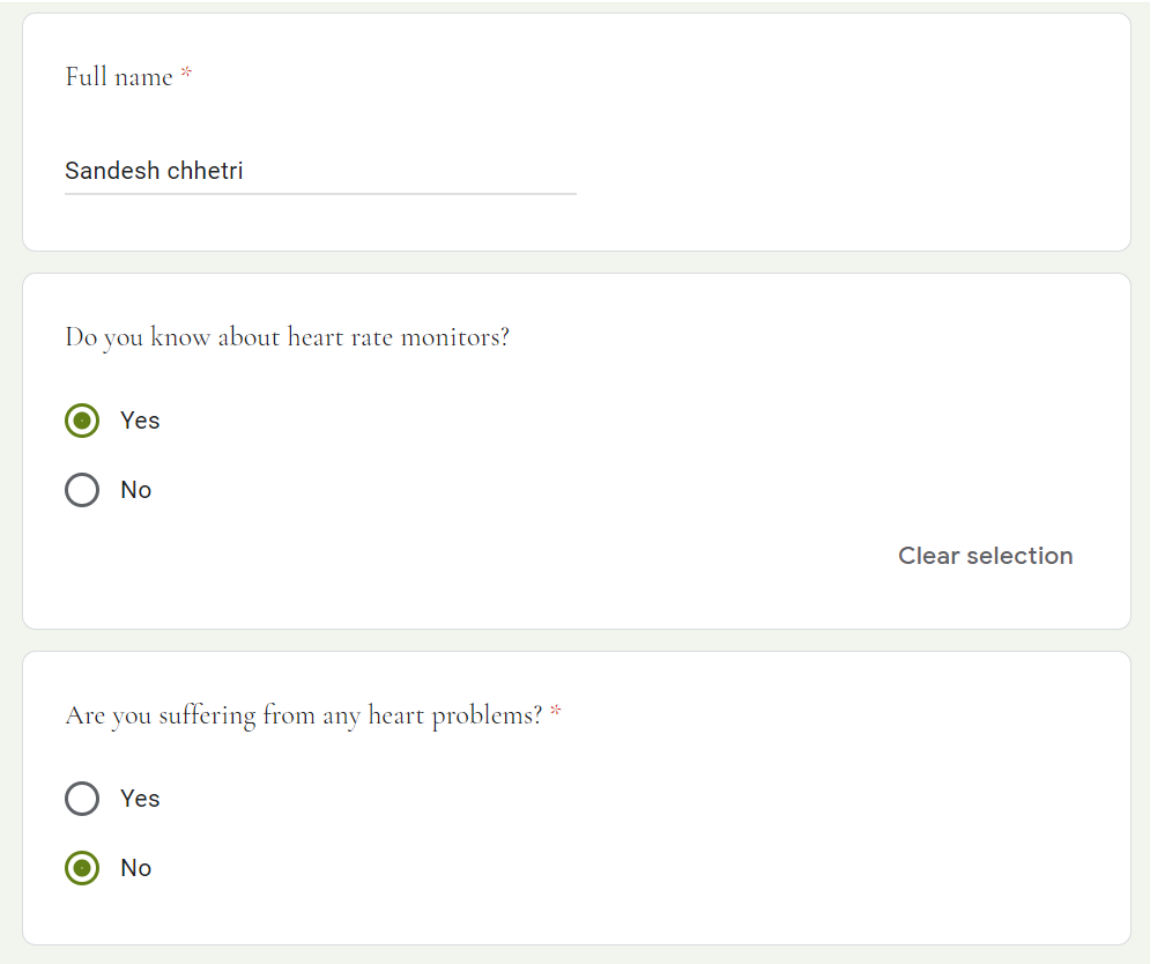

Figure 68 Filled Pre-Survey form 2

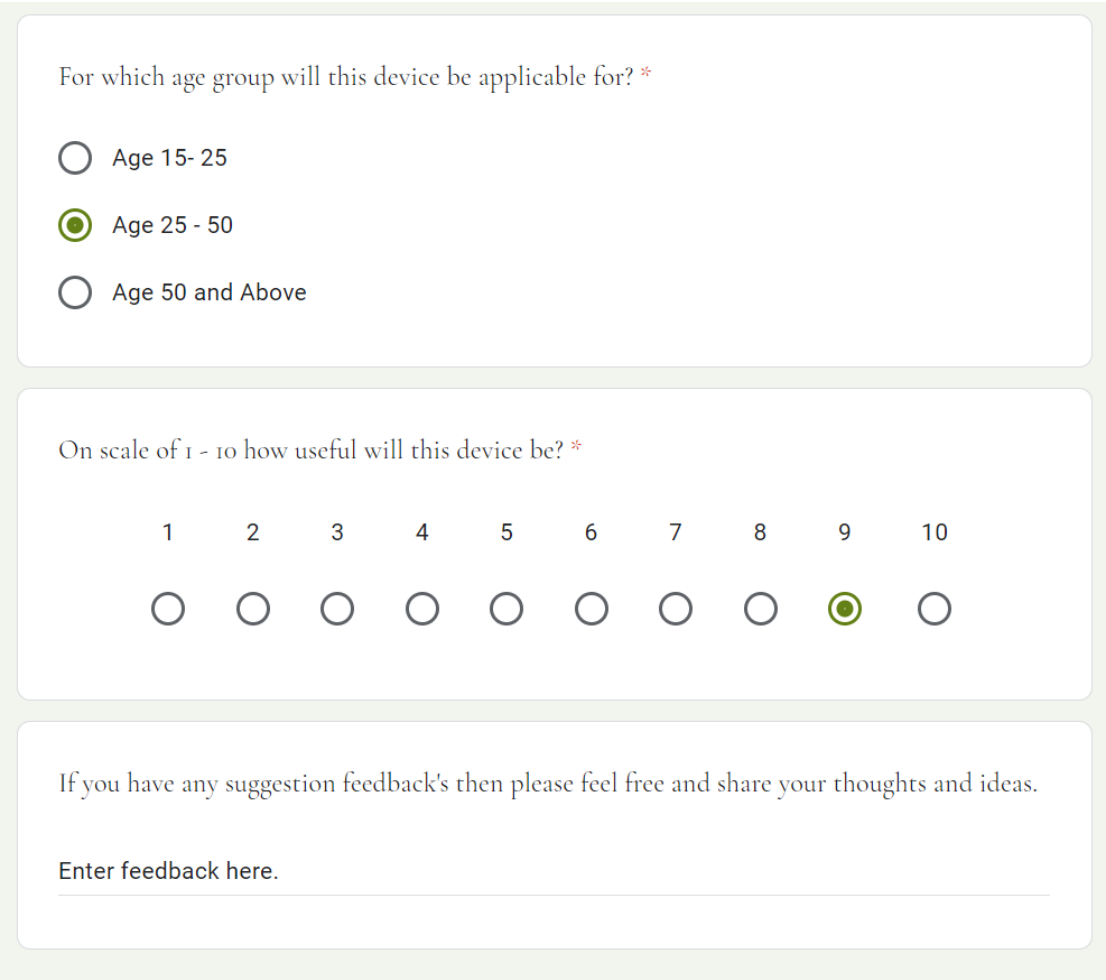

Figure 69 Filled Pre-Survey form 3

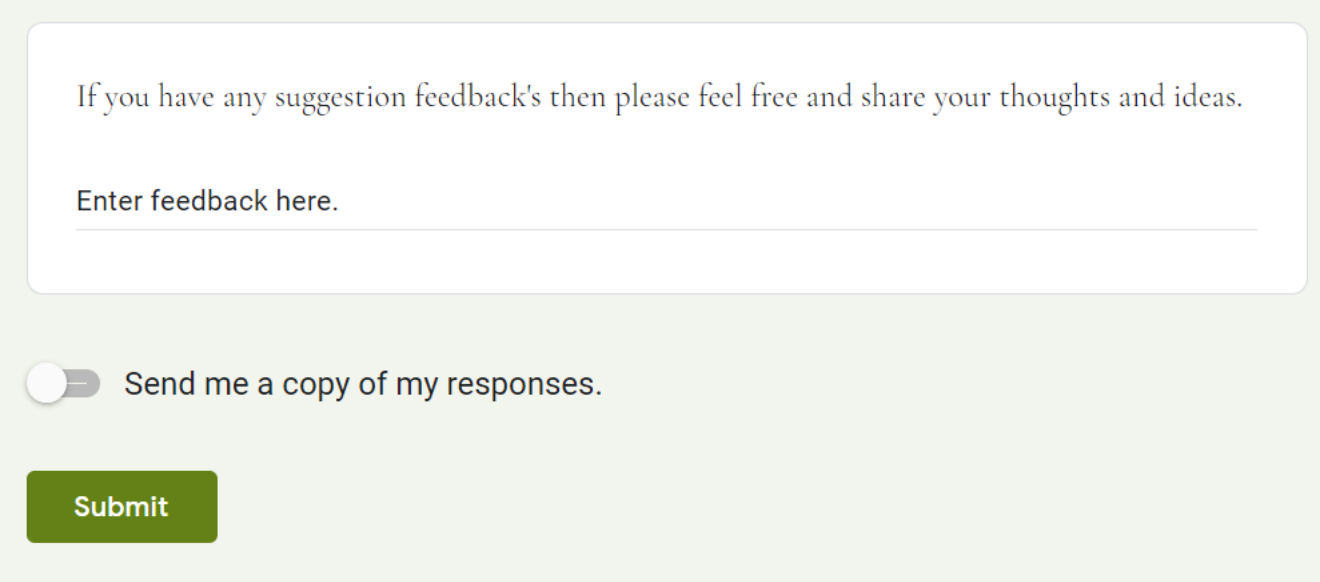

Figure 70 Filled Pre-Survey form 4

7.1.3 Pre-Survey results

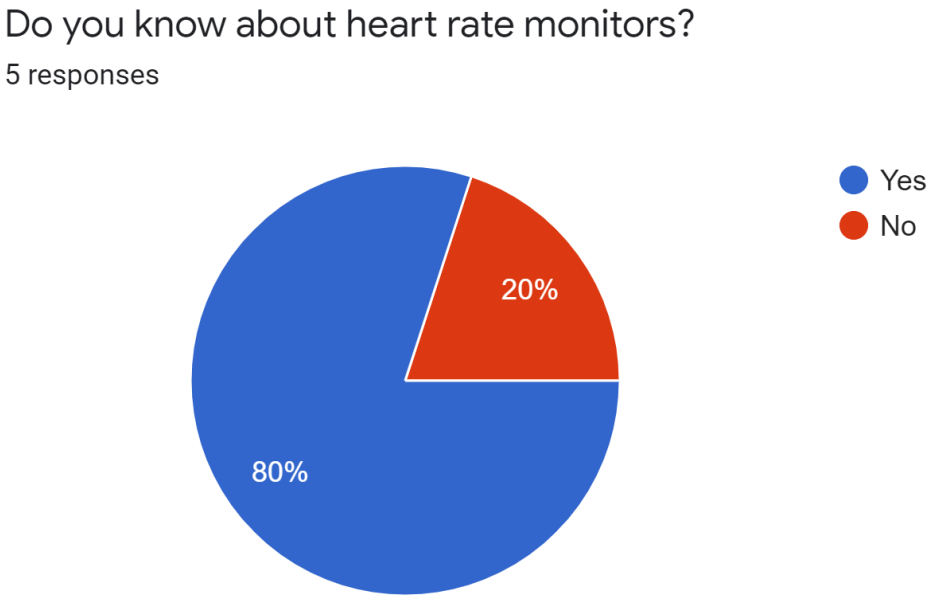

Figure 71 Pre-Survey results 1

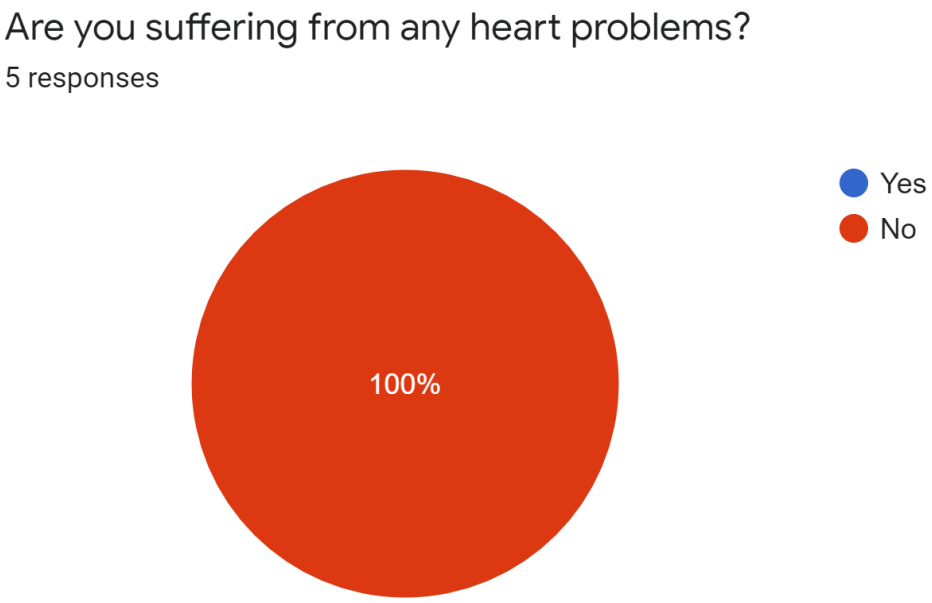

Figure 72 Pre-Survey results 2

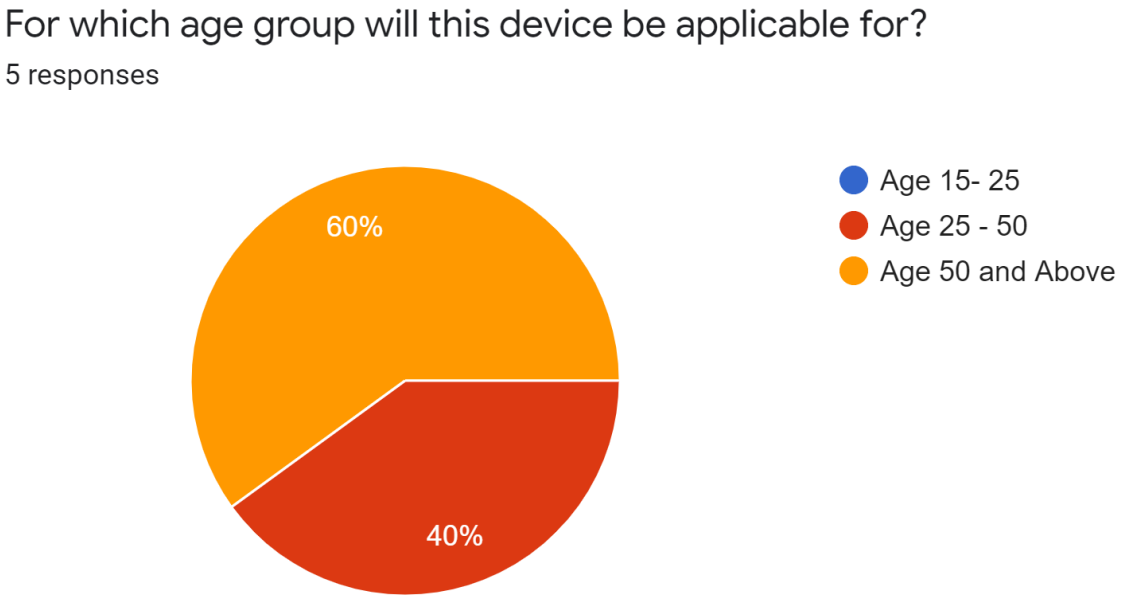

Figure 73 Pre-Survey results 3

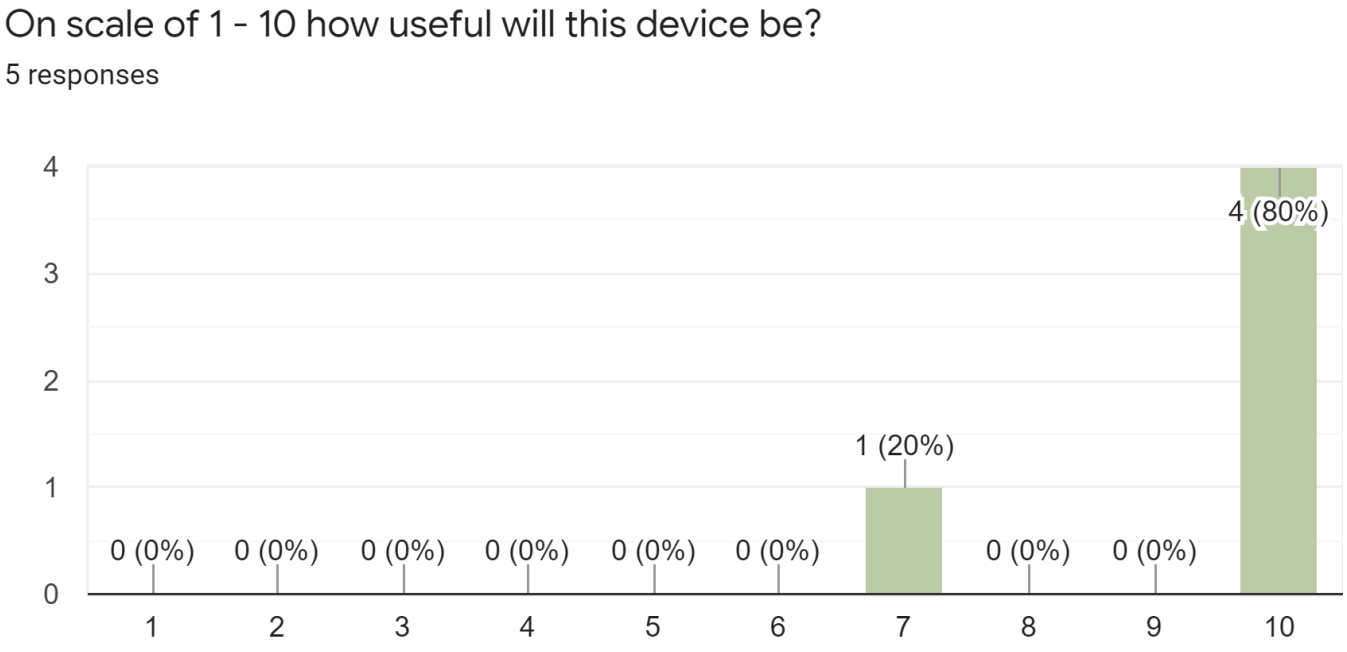

Figure 74 Pre-Survey results 4

7.2 Appendix B: Post-Survey

7.2.1 Post-Survey form

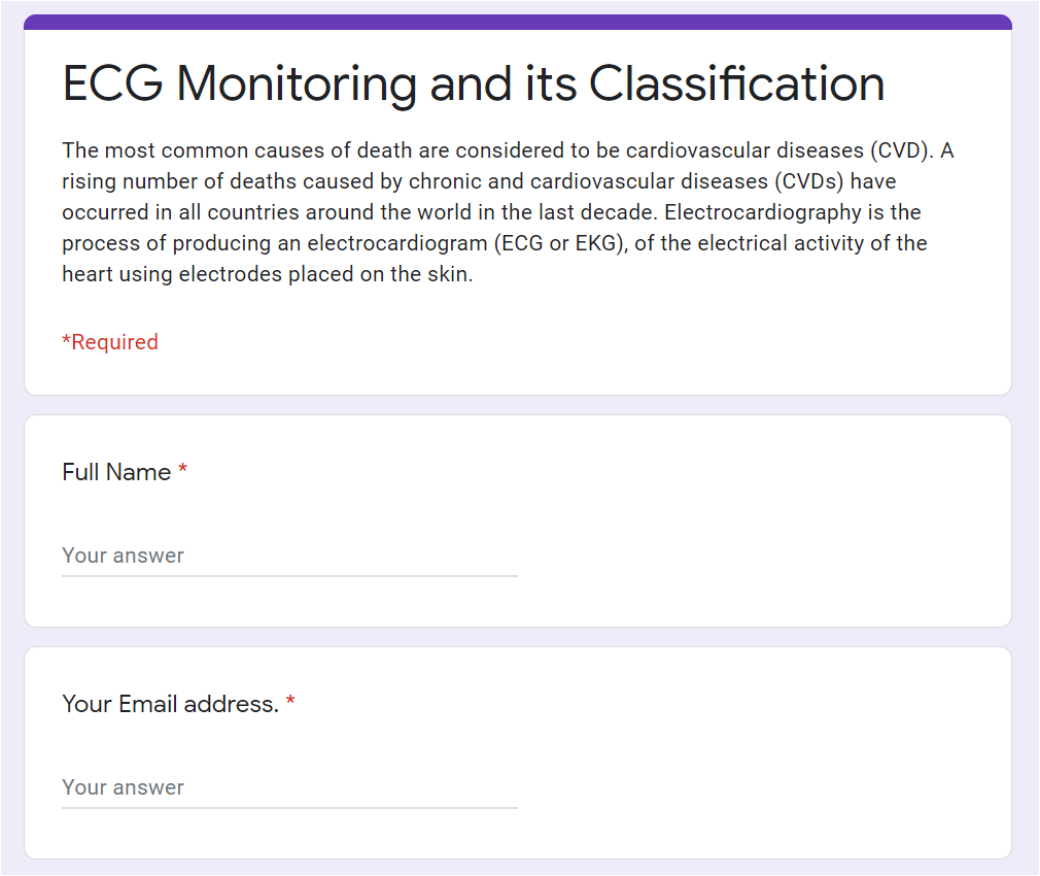

Figure 75 Post-Survey form 1

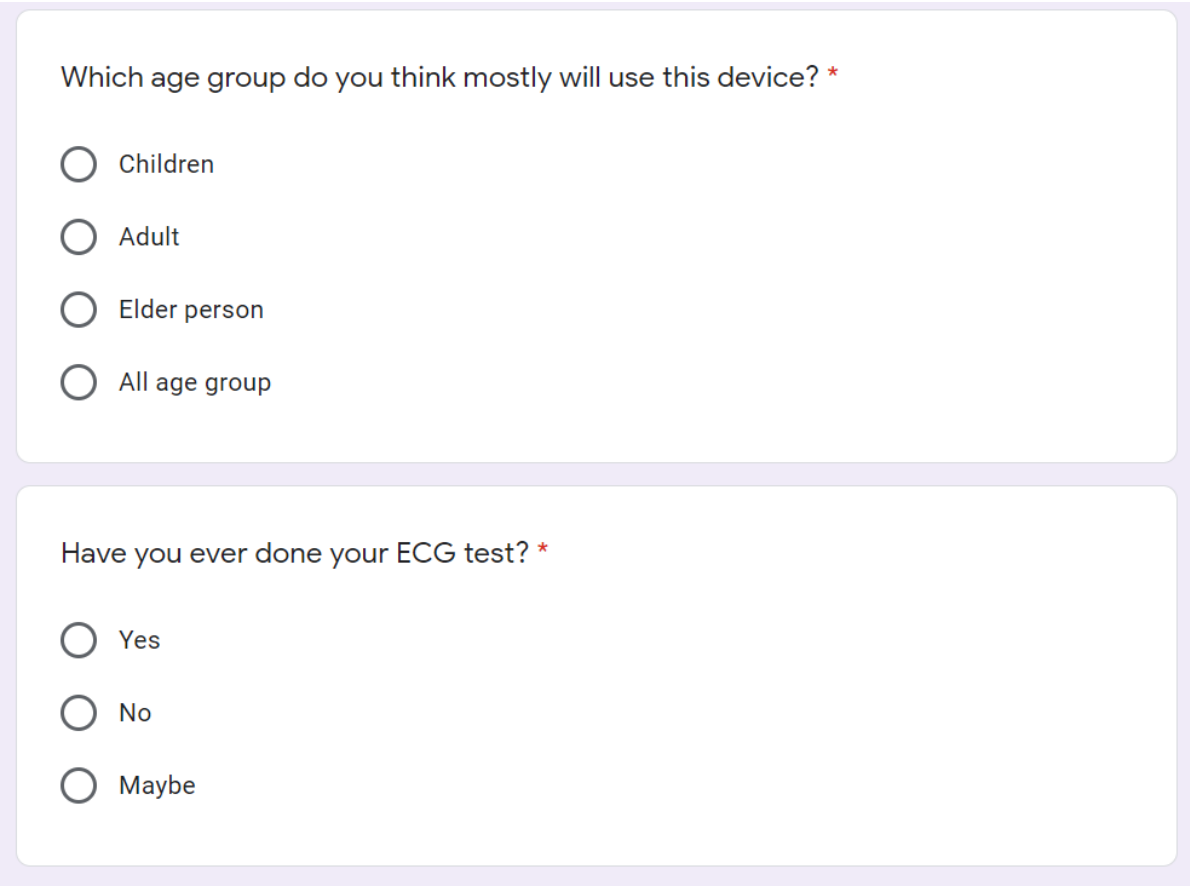

Figure 76 Post-Survey form 2

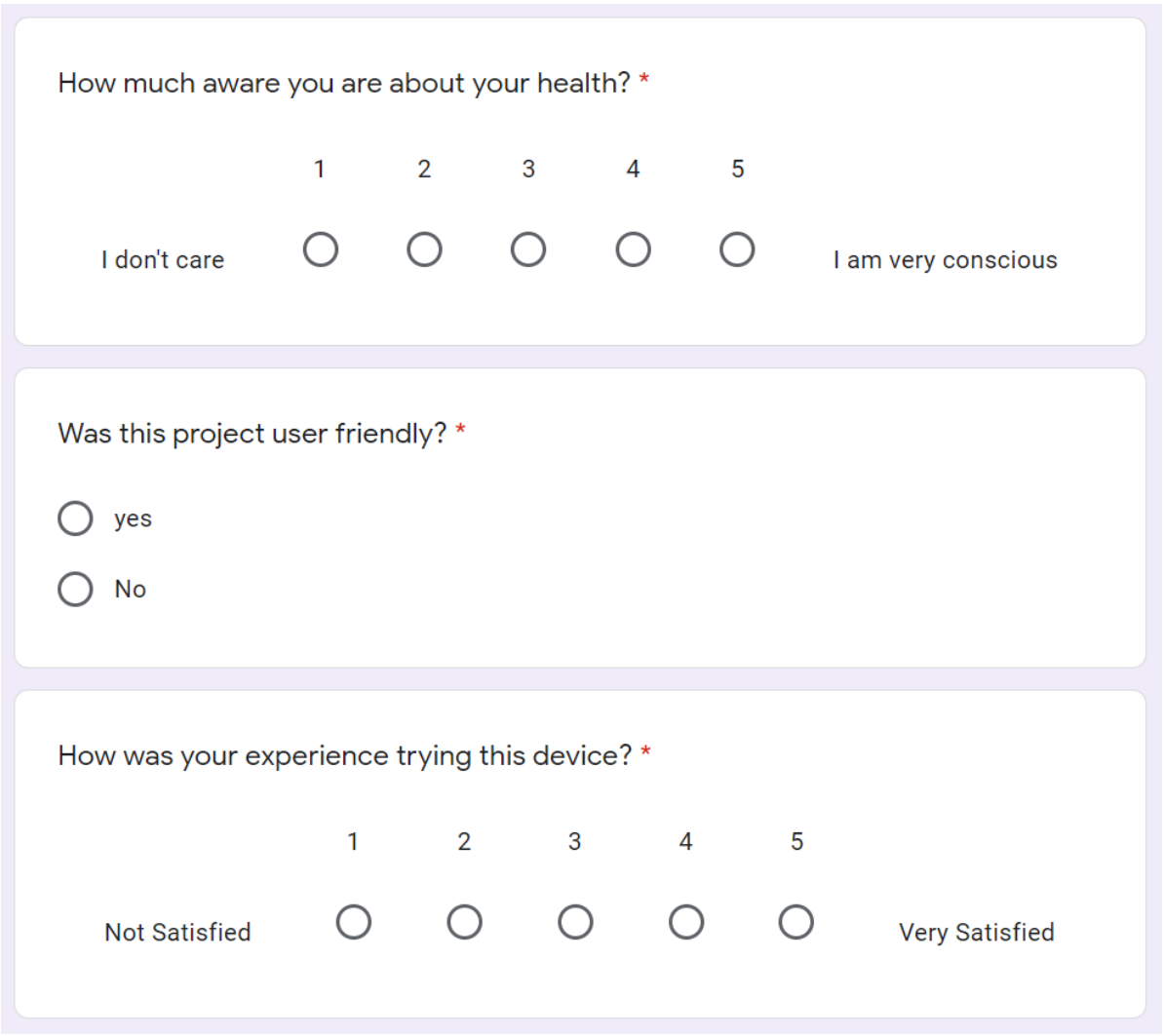

Figure 77 Post-Survey form 3

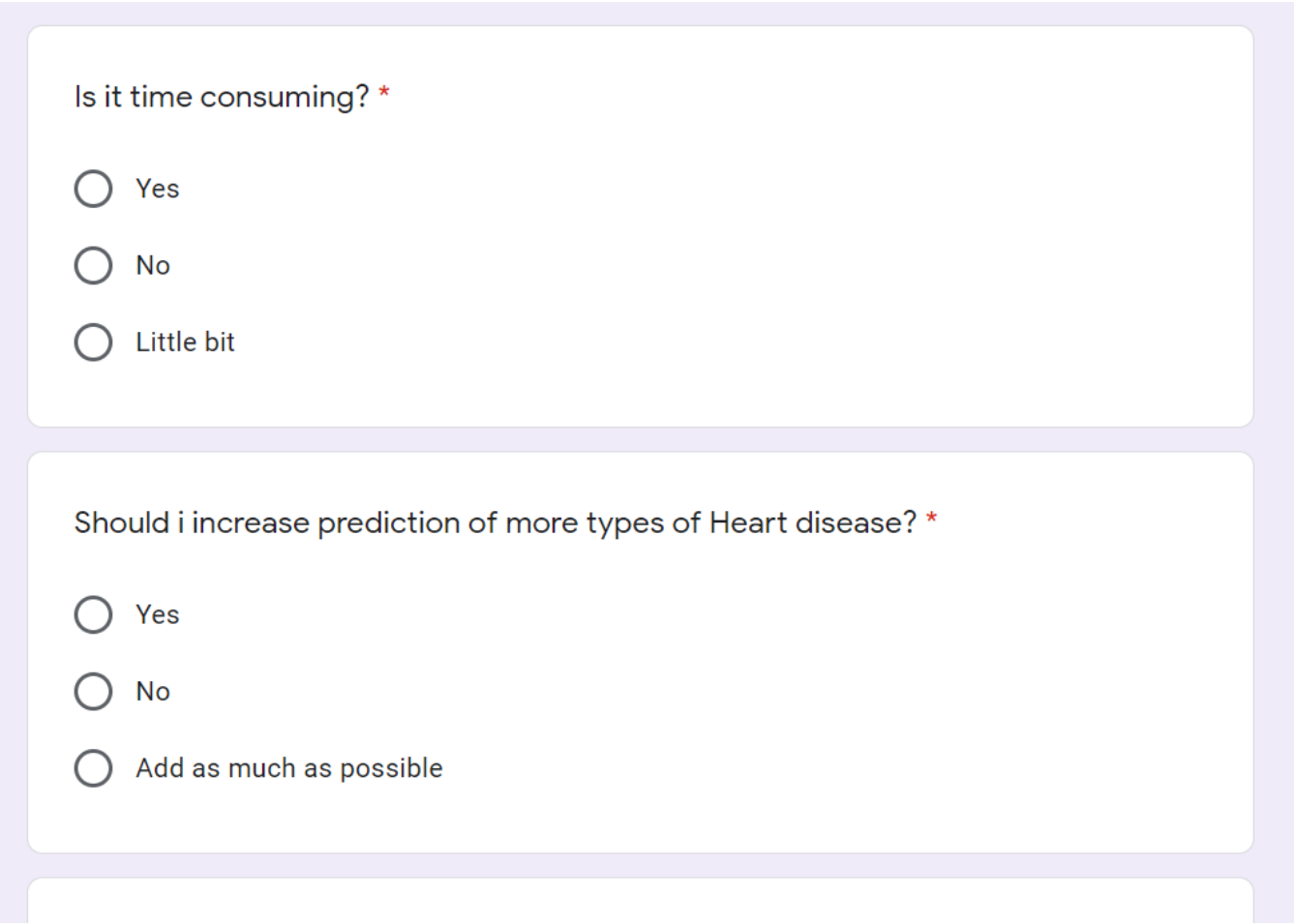

Figure 78 Post-Survey form 4

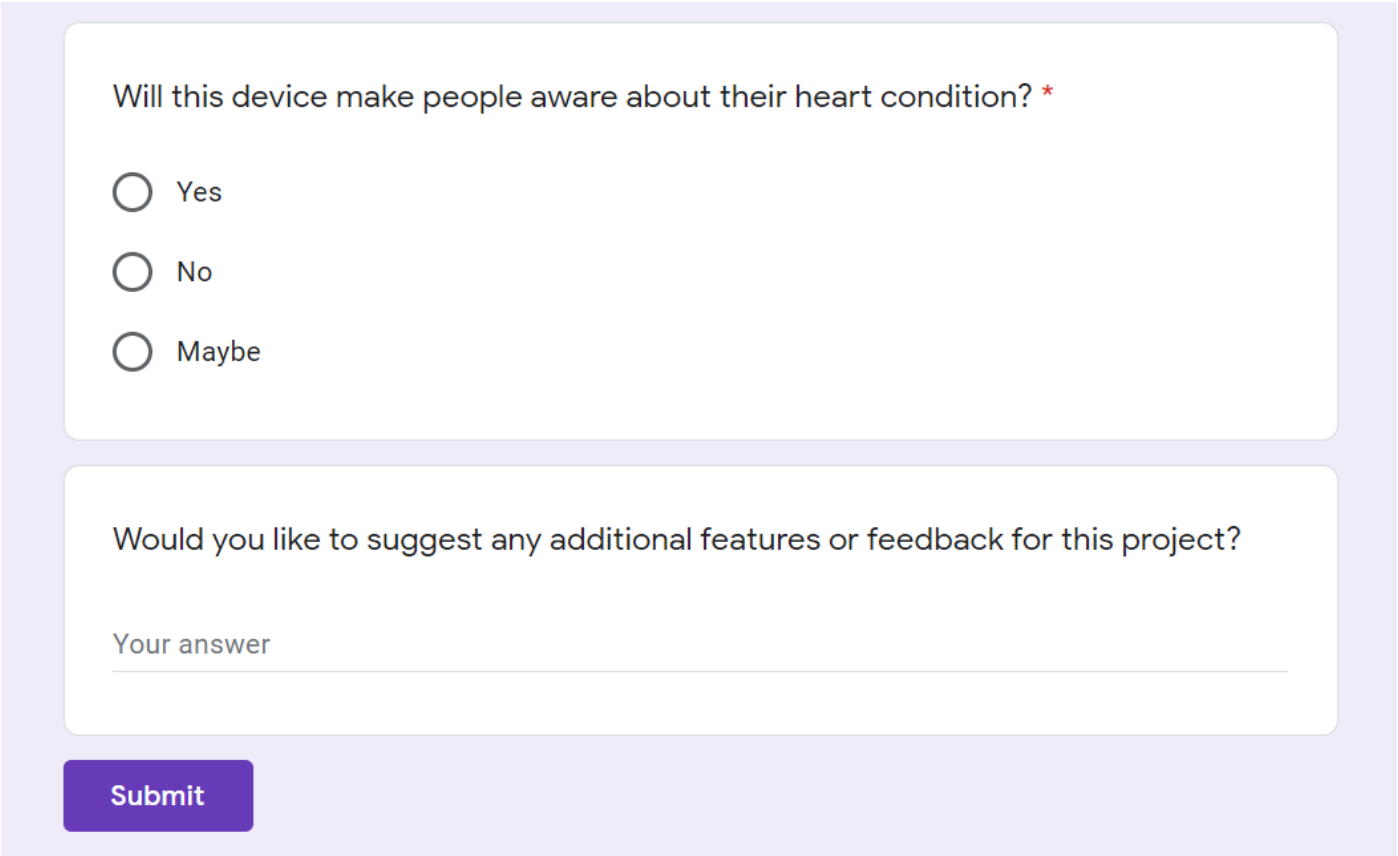

Figure 79 Post-Survey form 5

7.2.2 Sample of filled Post-Survey forms

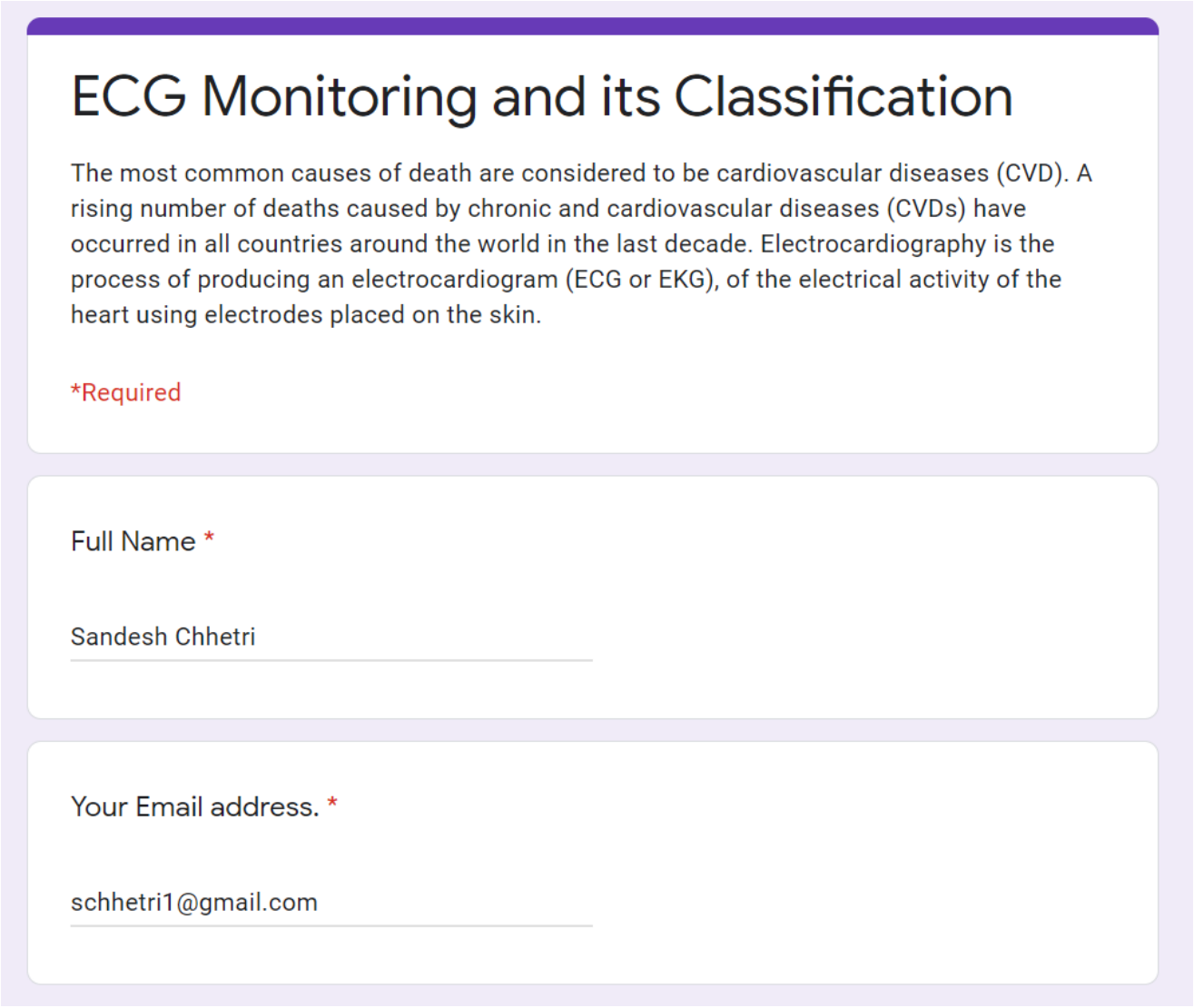

Figure 80 Filled Post-Survey form 1

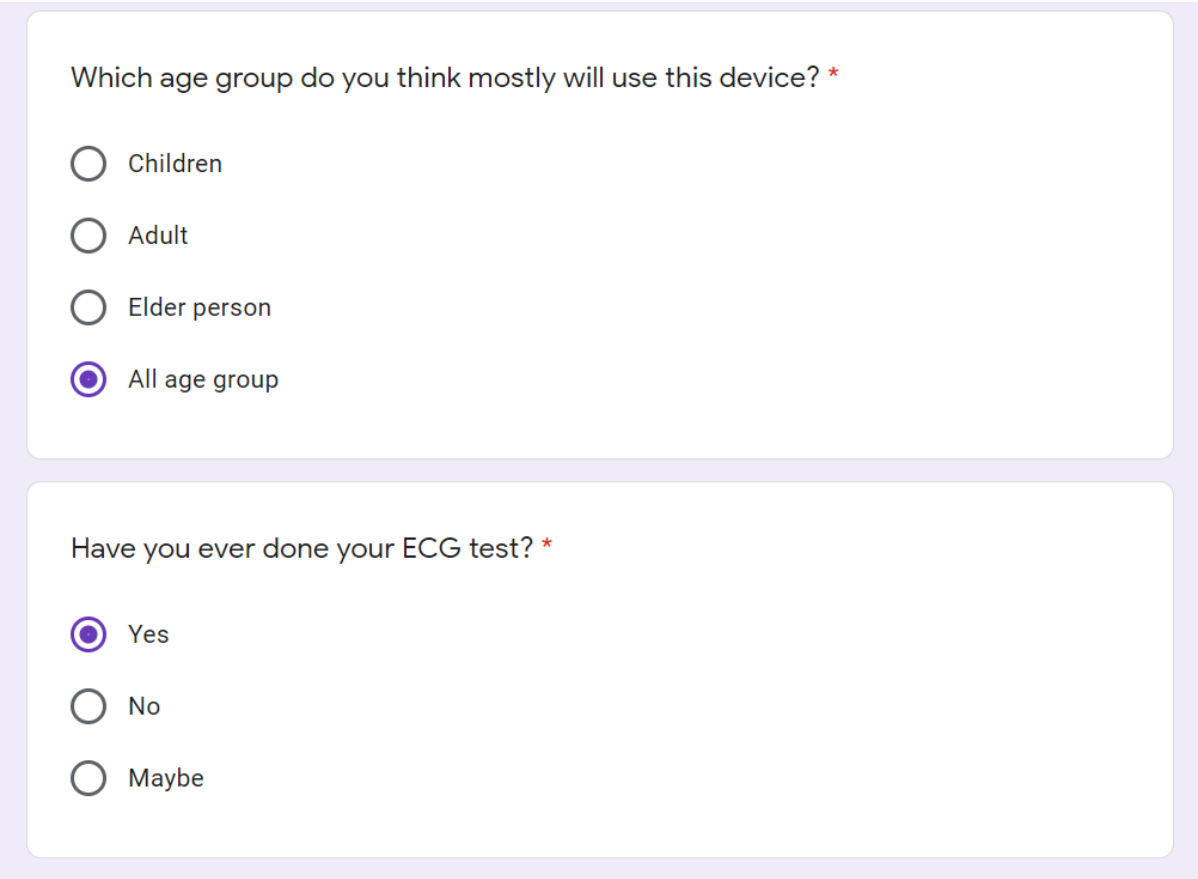

Figure 81 Filled Post-Survey form 2

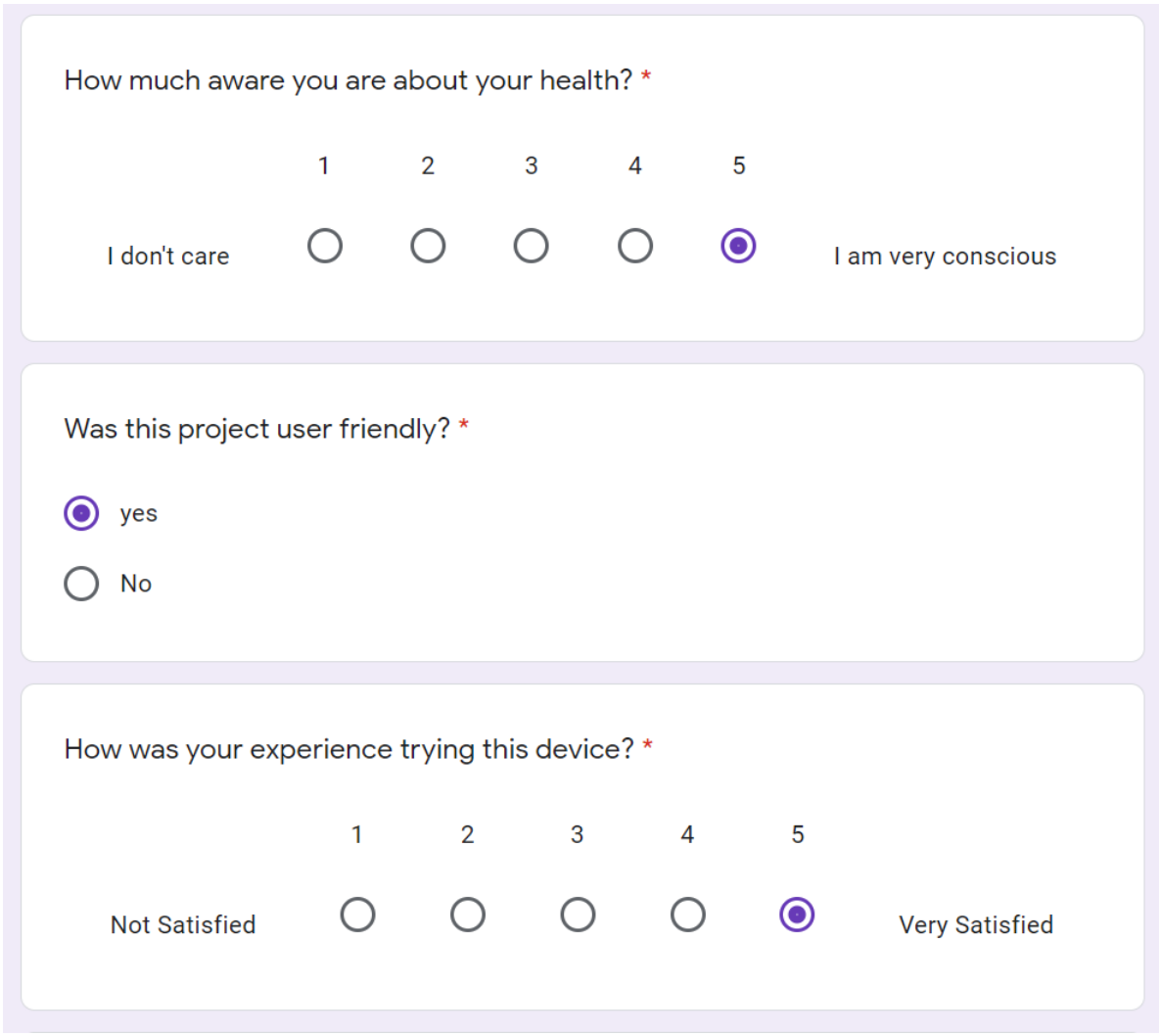

Figure 82 Filled Post-Survey form 3

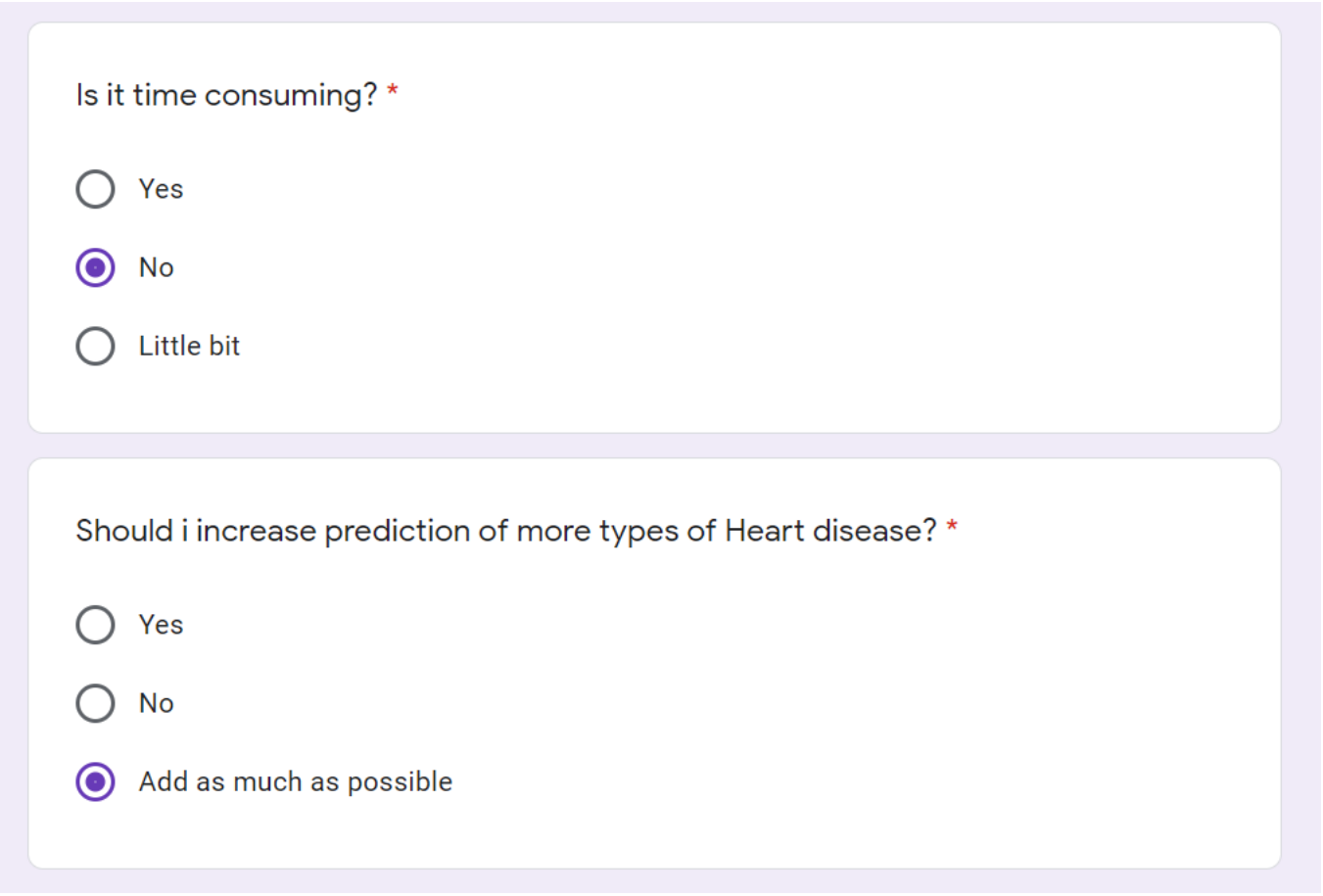

Figure 83 Filled Post-Survey form 4

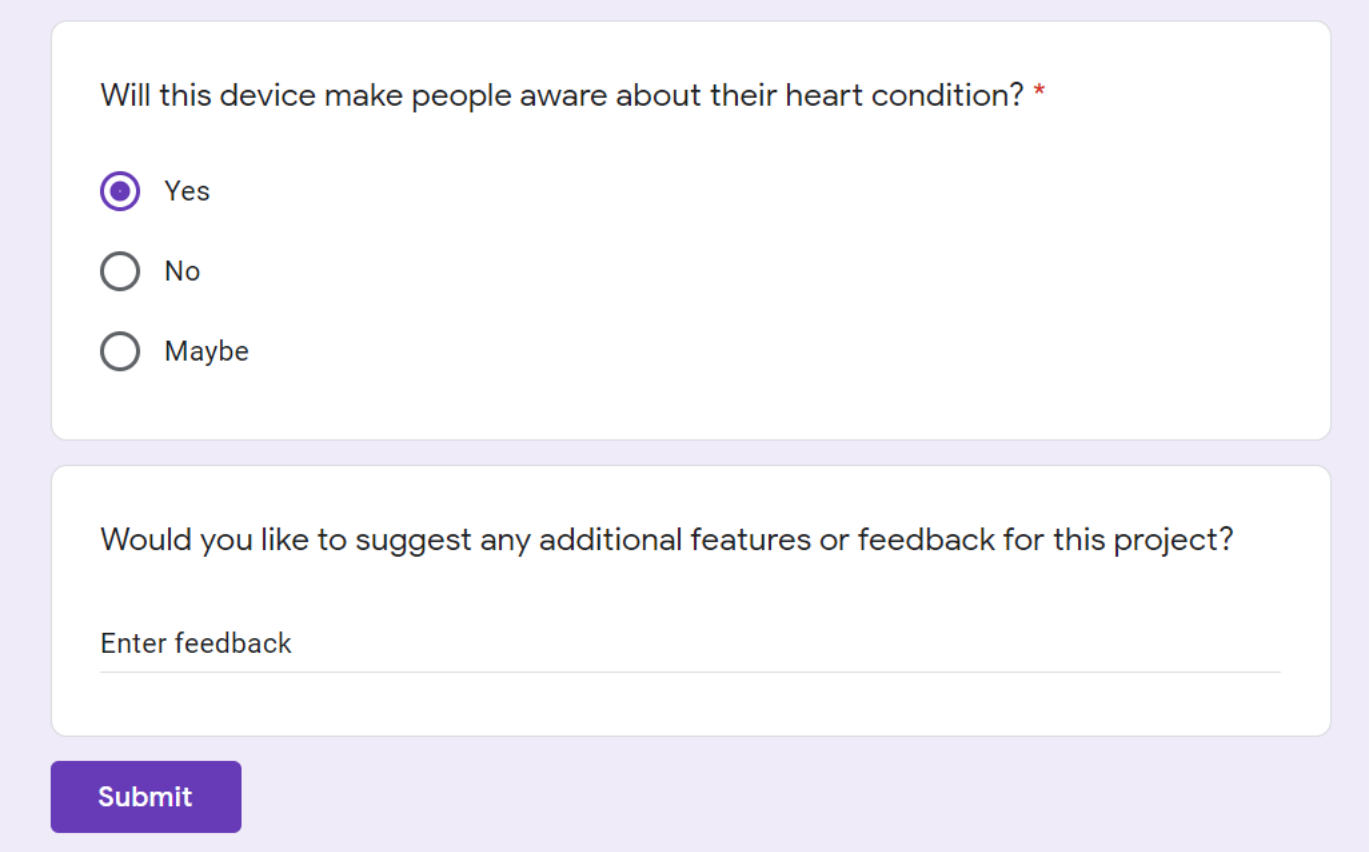

Figure 84 Filled Post-Survey form 5

7.2.3 Post-Survey result

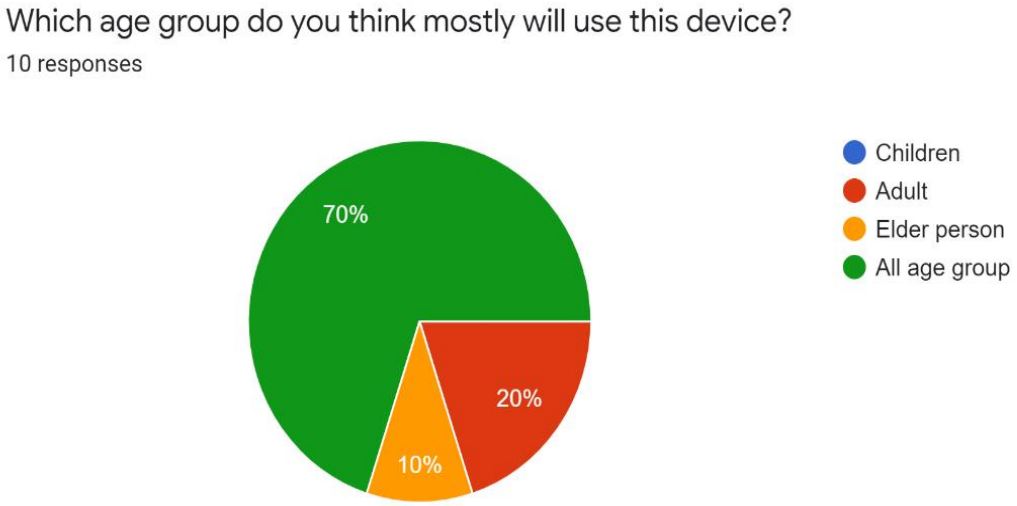

Figure 85 Post-Survey result 1

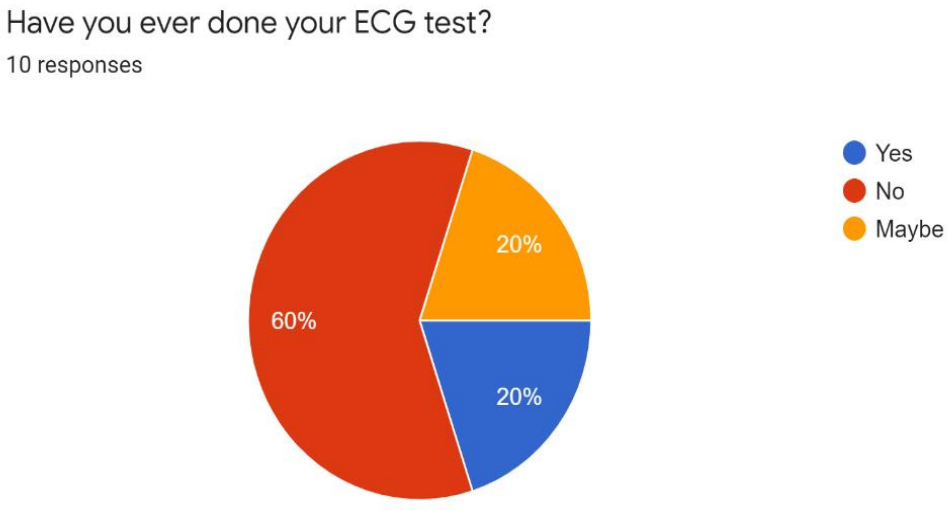

Figure 86 Post-Survey result 2

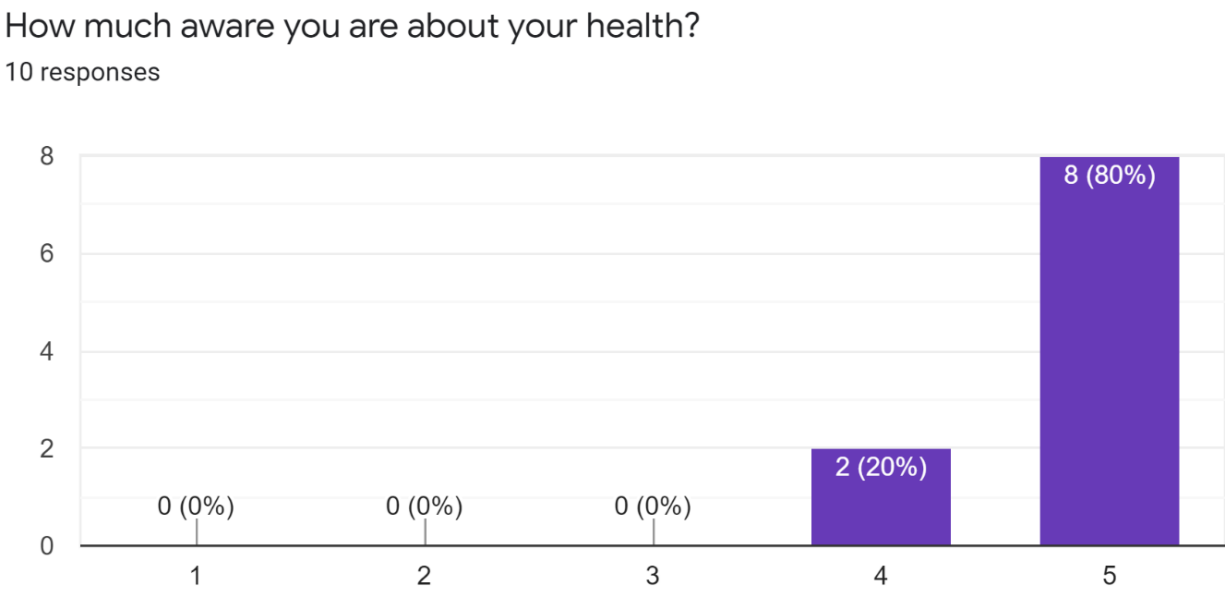

Figure 87 Post-Survey result 3

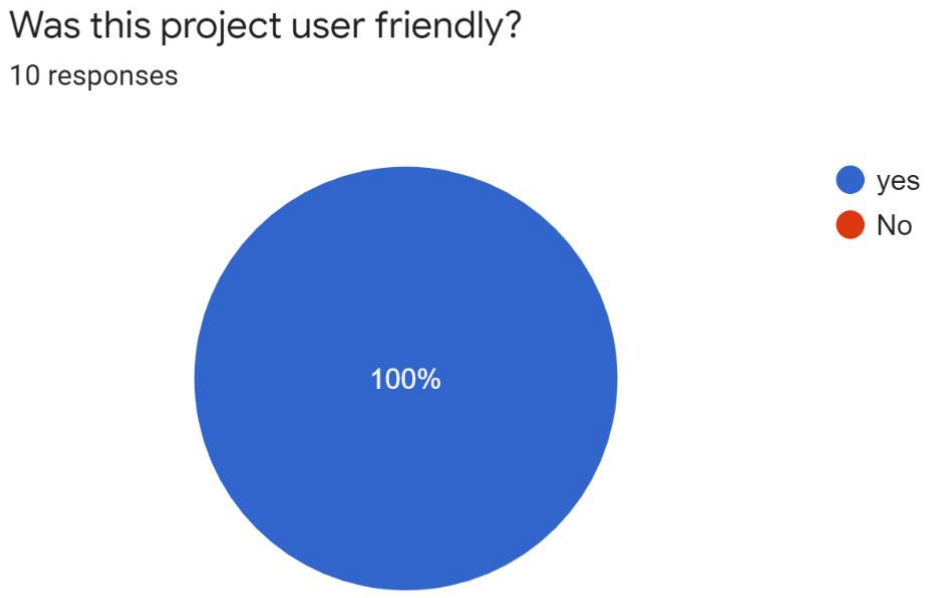

Figure 88 Post-Survey result 4

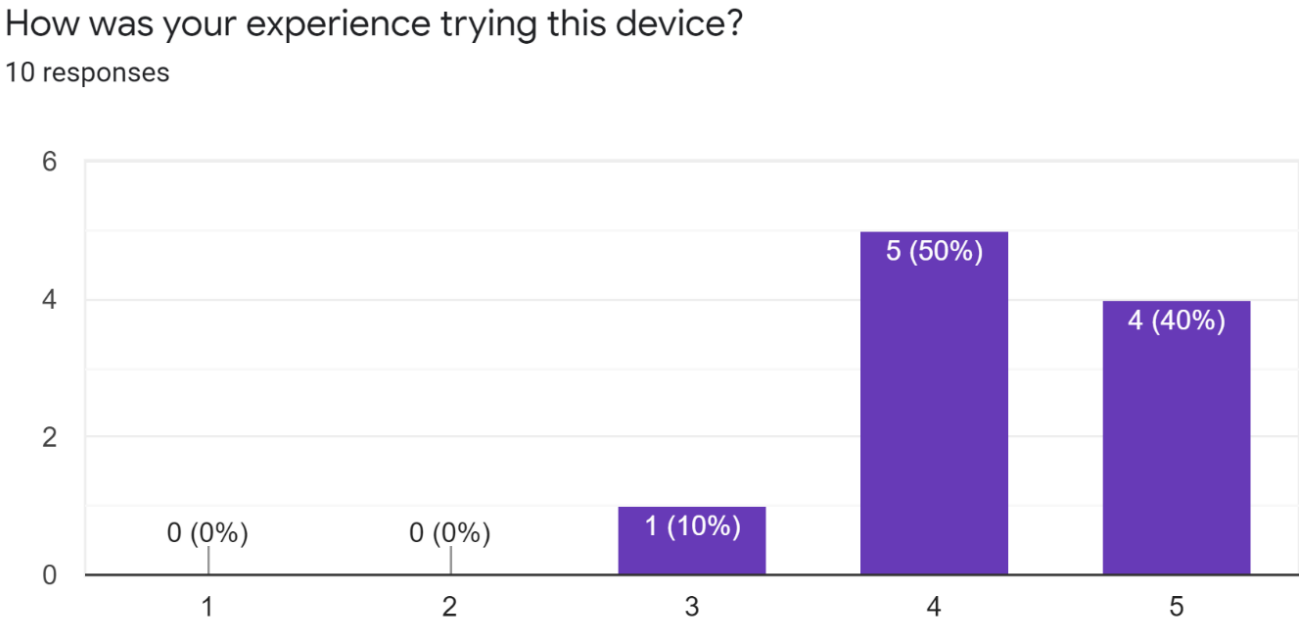

Figure 89 Post-Survey result 5

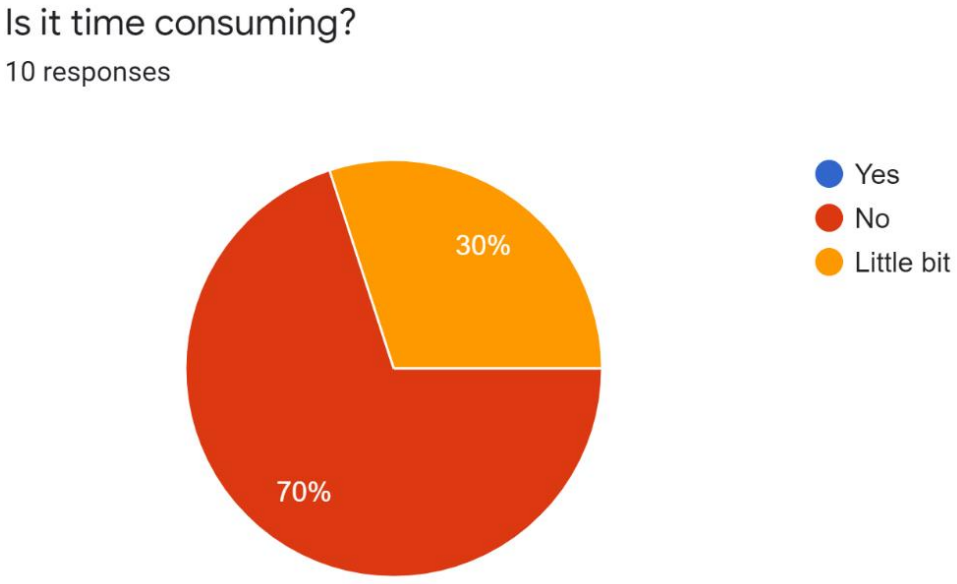

Figure 90 Post-Survey result 6

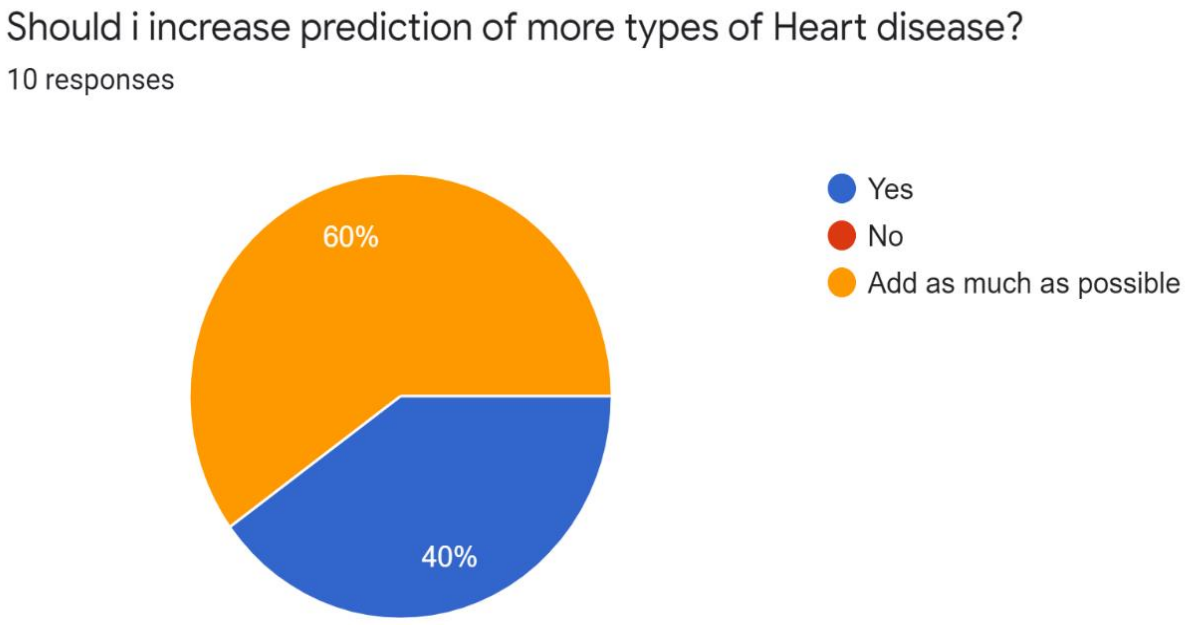

Figure 91 Post-Survey result 7

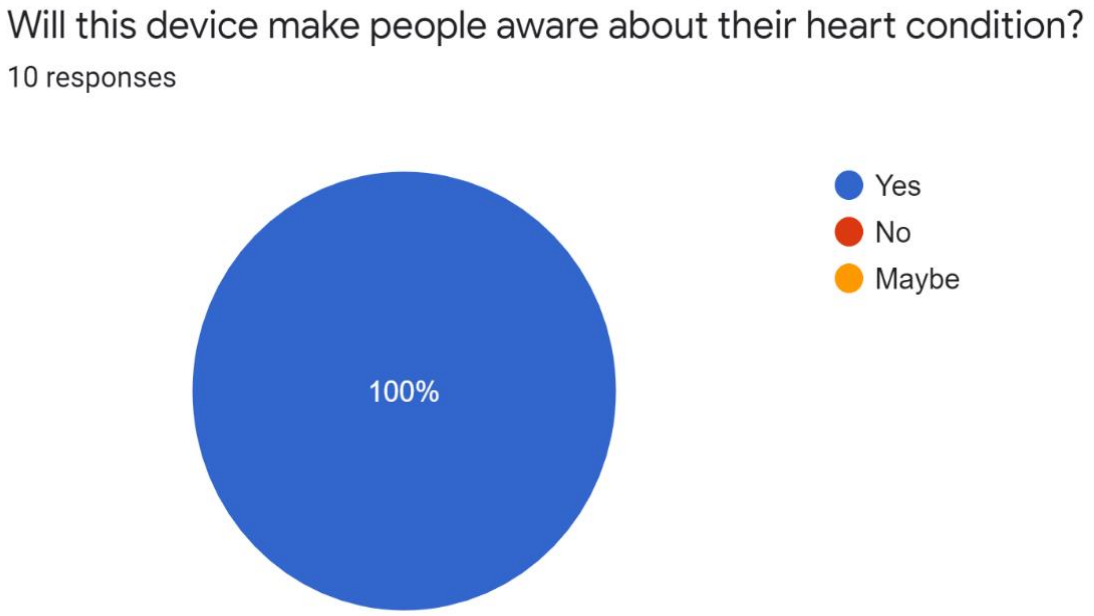

Figure 92 Post-Survey result 8

7.3 Appendix C: Sample Codes

7.3.1 Sample code of the UI

```
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>ECG result</title>
    <link href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css" rel="stylesheet">
    <script src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"></script>
    <script src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></script>
    <script src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
    <link href="{{ url_for('static', filename='css/main.css') }}" rel="stylesheet">
</head>

<body>
    <nav class="navbar navbar-dark bg-dark">
        <div class="container">
            <a class="navbar-brand" href="#">Stay Healthy, Stay Happy by- Sandesh Chhetri</a>
            <button class="btn btn-outline-secondary my-2 my-sm-0" type="submit">Help</button>
        </div>
    </nav>
    <div class="container">
        <div id="content" style="margin-top:2em">{% block content %}{% endblock %}</div>
    </div>
</body>

<footer>
    <script src="{{ url_for('static', filename='js/main.js') }}" type="text/javascript"></script>
</footer>

</html>

```

Figure 93 Base html code

```
{% extends "base.html" %} {% block content %}

<h2>ECG Arrhythmia Classifier</h2>

<div>
    <form id="upload-file" method="post" enctype="multipart/form-data">
        <label for="imageUpload" class="upload-label">
            Choose...
        </label>
        <input type="file" name="file" id="imageUpload" accept=".csv">
    </form>

    <div class="image-section" style="display:none;">
        <div>
            <div>
            </div>
        </div>
        <div>
            <button type="button" class="btn btn-primary btn-lg " id="btn-predict">Predict!</button>
        </div>
    </div>

    <div class="loader" style="display:none;"></div>

    <h3 id="result">
        <span> </span>
    </h3>

</div>

{% endblock %}

```

Figure 94 Index html code

```css
.img-preview {
    width: 256px;
    height: 256px;
    position: relative;
    border: 5px solid #F8F8F8;
    box-shadow: 0px 2px 4px 0px rgba(0, 0, 0, 0.1);
    margin-top: 1em;
    margin-bottom: 1em;
}

.img-preview>div {
    width: 100%;
    height: 100%;
    background-size: 256px 256px;
    background-repeat: no-repeat;
    background-position: center;
}

input[type="file"] {
    display: none;
}

.upload-label{
    display: inline-block;
    padding: 12px 30px;
    background: #39D2B4;
    color: #fff;
    font-size: 1em;
    transition: all .4s;
    cursor: pointer;
}

.upload-label:hover{
    background: #34495E;
    color: #39D2B4;
}

.loader {
    border: 8px solid #f3f3f3; /* Light grey */
    border-top: 8px solid #3498db; /* Blue */
    border-radius: 50%;
    width: 50px;
    height: 50px;
    animation: spin 1s linear infinite;
}

@keyframes spin {
    0% { transform: rotate(0deg); }
    100% { transform: rotate(360deg); }
}
```

Figure 95 CSS code

```javascript
$(document).ready(function () {
    // Init
    $('.image-section').hide();
    $('.loader').hide();
    $('#result').hide();

    // Upload Preview
    function readURL(input) {
        if (input.files && input.files[0]) {
            var reader = new FileReader();
            reader.onload = function (e) {
                $('#imagePreview').css('background-image', 'url(' + e.target.result + ')');
                $('#imagePreview').hide();
                $('#imagePreview').fadeIn(650);
            }
            reader.readAsDataURL(input.files[0]);
        }
    }
    $("#imageUpload").change(function () {
        $('.image-section').show();
        $('#btn-predict').show();
        $('#result').text('');
        $('#result').hide();
        readURL(this);
    });

    // Predict
    $('#btn-predict').click(function () {
        var form_data = new FormData($('#upload-file')[0]);

        // Show loading animation
        $(this).hide();
        $('.loader').show();

        // Make prediction by calling api /predict
        $.ajax({
            type: 'POST',
            url: '/predict',
            data: form_data,
            contentType: false,
            cache: false,
            processData: false,
            async: true,
            success: function (data) {
                // Get and display the result
                $('.loader').hide();
                $('#result').fadeIn(600);
                $('#result').text(' Result:  ' + data);
                console.log('Success!');
            },
        });
    });

});

```

Figure 96 JavaScript code

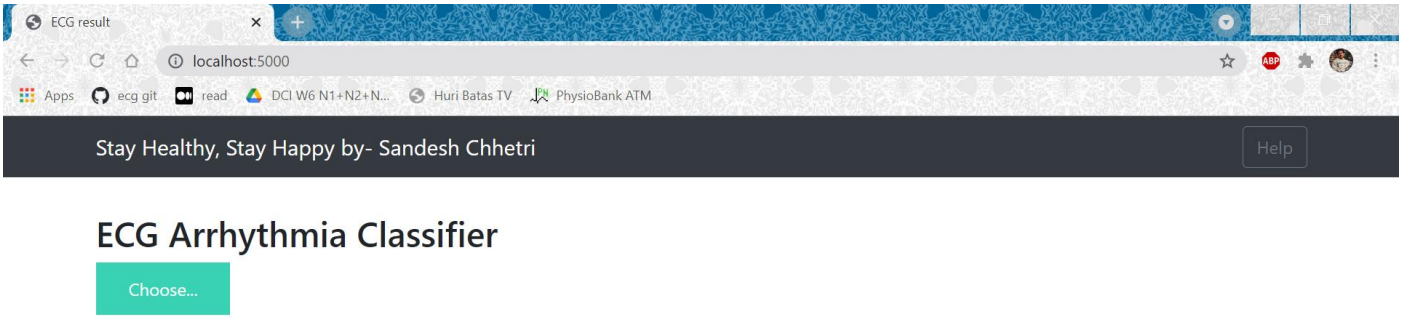

Figure 97 Webpage

7.4 Appendix D: Designs

7.4.1 Gantt Chart

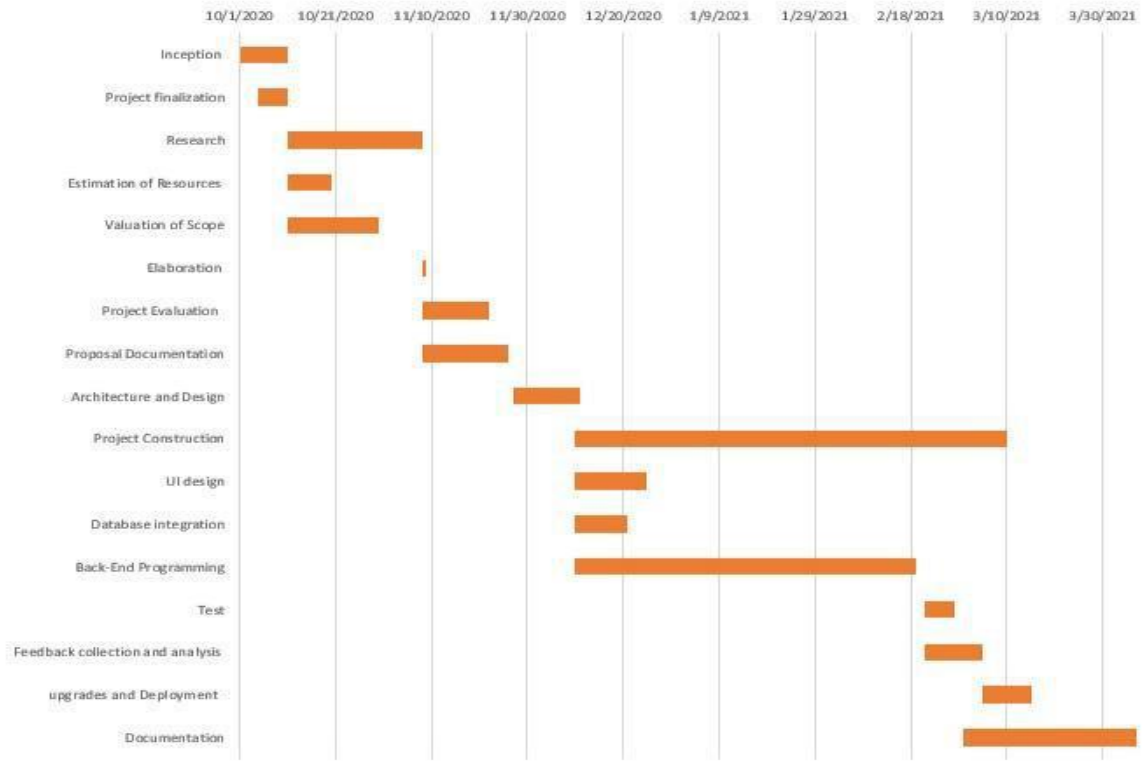

Figure 98 Gantt Chart

7.4.2 Work Breakdown Structure

No	Work and Description	Start Date	End Date
1.	Web searching and collecting all data required for design of the project.	11/28/2020	12/04/2020
2.	Hardware architecture design with all mechanism.	12/05/2020	12/20/2020
3.	Arrangement of hardware and creating hardware according to the proposed architecture design.	12/21/2020	1/20/2021
4.	Software Architecture design with all required features.	1/21/2020	2/10/2021
5.	Software prototype and software creation according to the software architecture design	2/11/2020	3/10/2021
6.	Testing of hardware and software created with various test cases.	3/11/2021	3/15/2021
7.	Updating the results functions according to suggestion from supervisors.	3/16/2021	3/20/2021
8.	Finalizing project with fulfilment of all required objectives.	3/16/2021	3/25/2021
9.	Documentation	3/26/2021	4/10/2021

Table 11 Work Breakdown Structure

7.4.3 Algorithms and Flowchart

Algorithm

Step 1: Start

Step 2: Input ECG signals.

Step 3: Process ECG signal as ECG image and save it.

Step 4: Data Augmentation

Step 5: Using CNN algorithm

Step 6: Classified result as output

Step 7: Stop

Flowchart

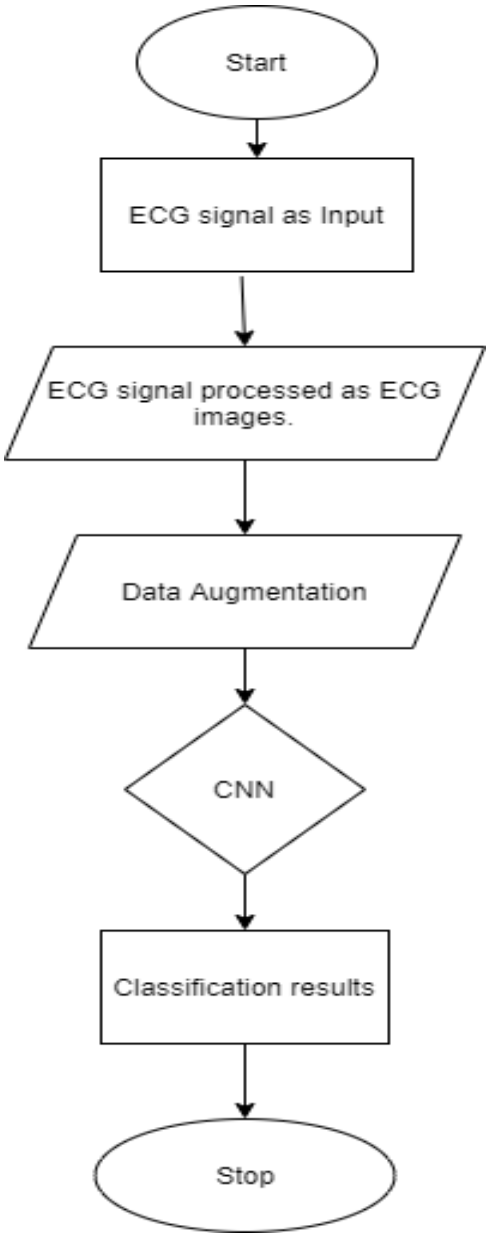

Figure 99 Flowchart of Project

7.4.4 Hardware Architecture

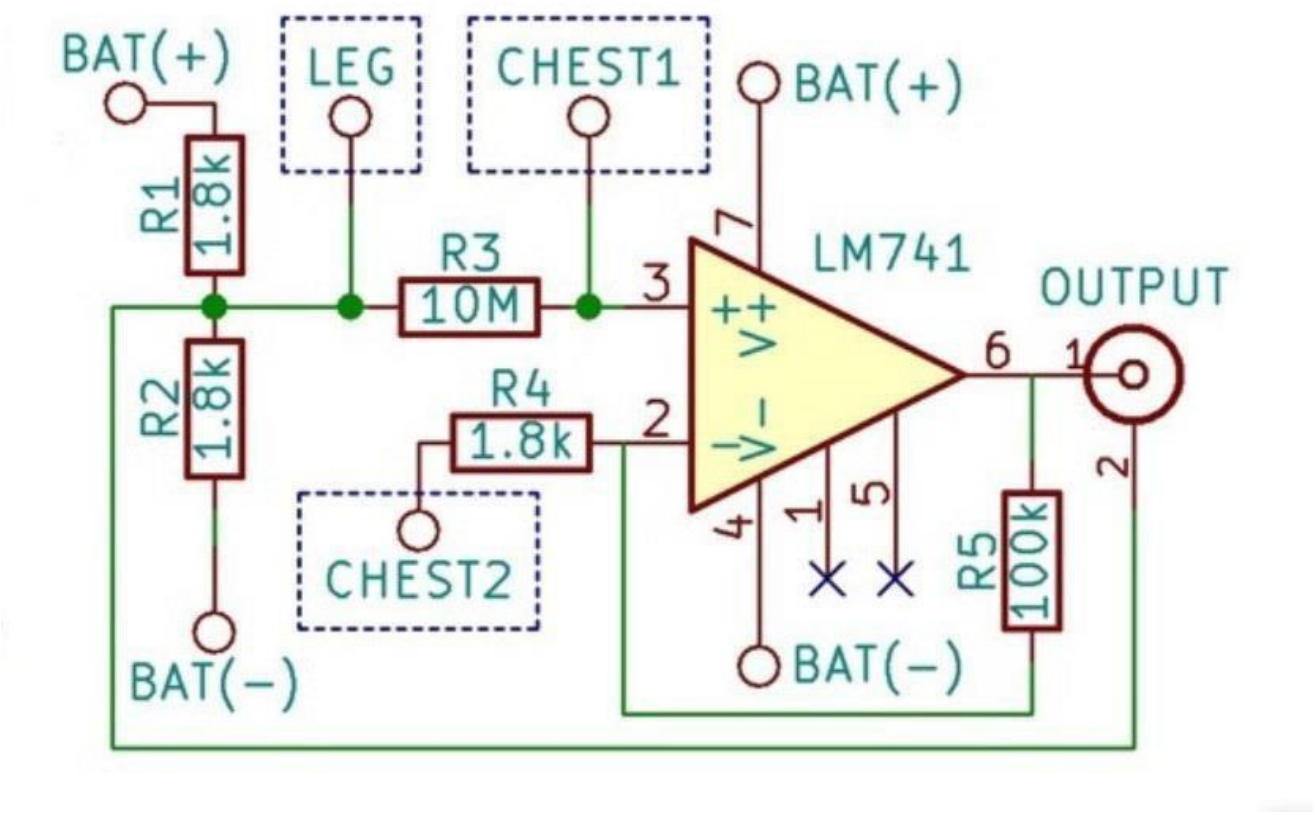

Figure 100 Hardware Architecture

7.4.5 Wireframe

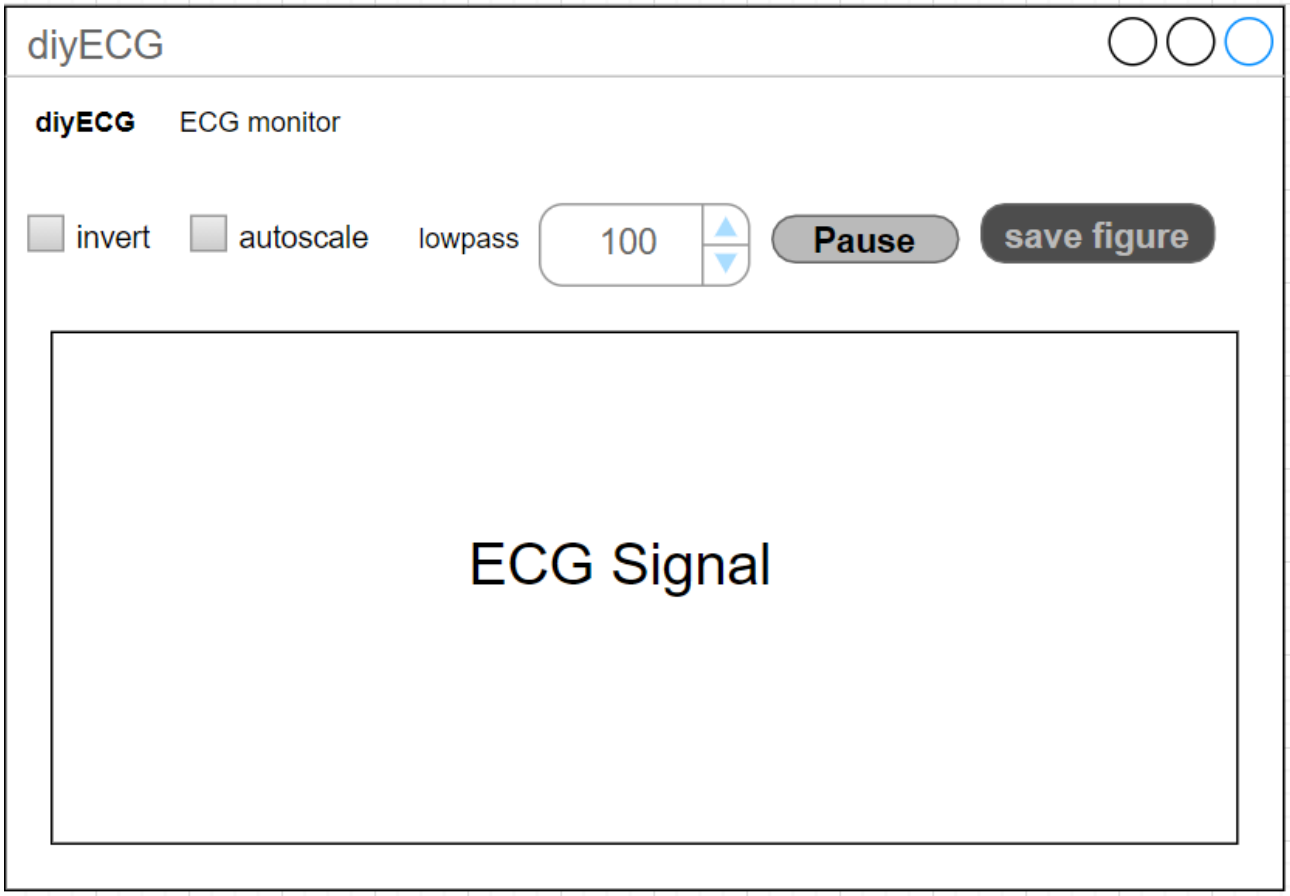

Figure 101 Wireframe of ECG data

Figure 102 Wireframe of webpage 1

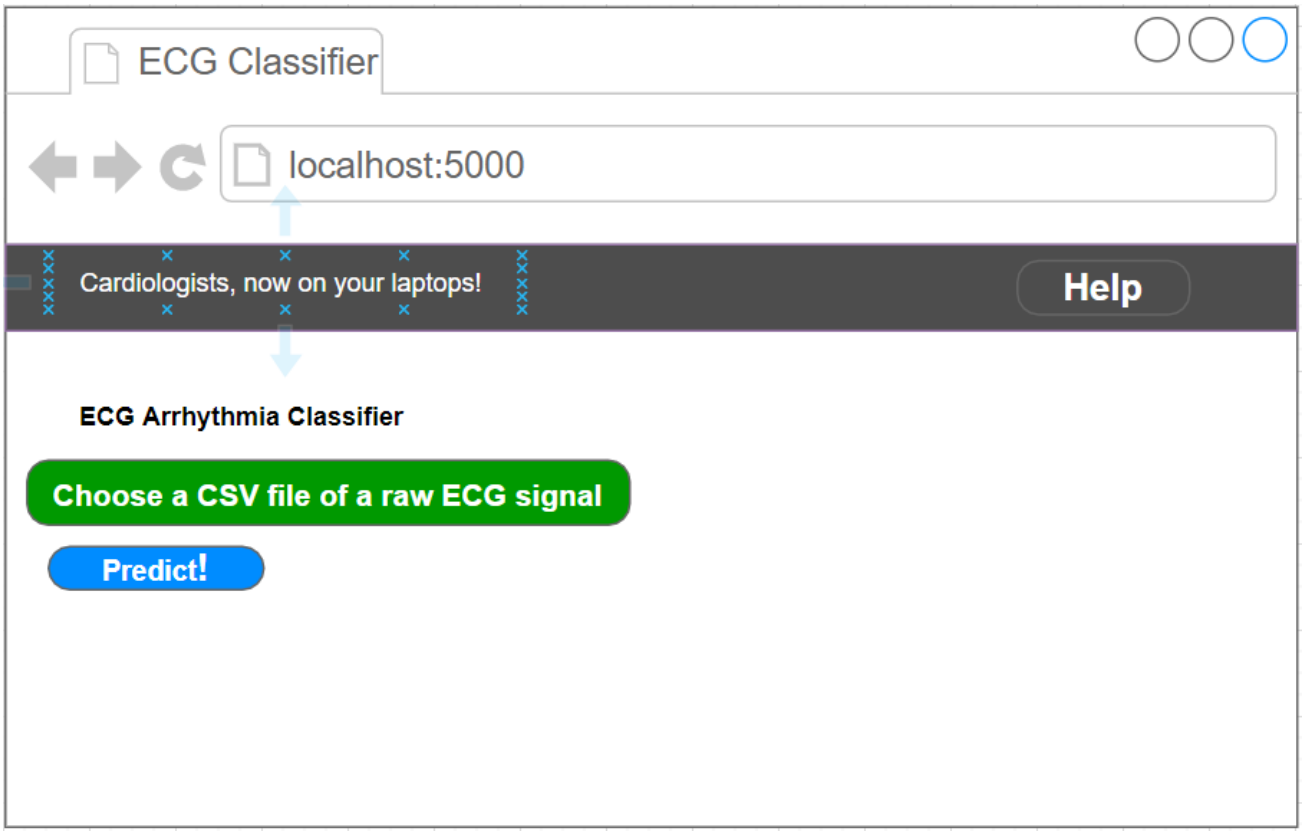

Figure 103 Wireframe of webpage 2

7.5 Appendix E: Screenshots of the system

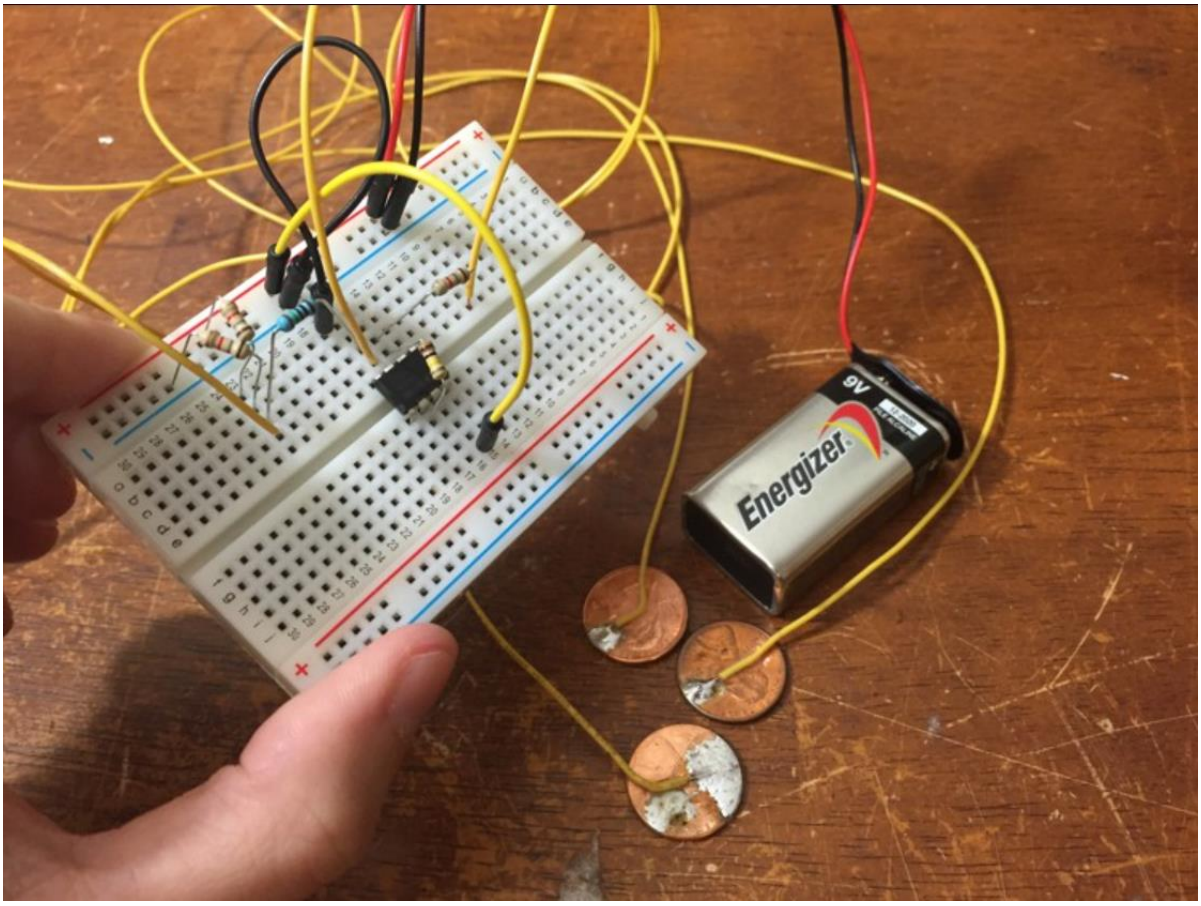

Figure 104 Hardware configuration

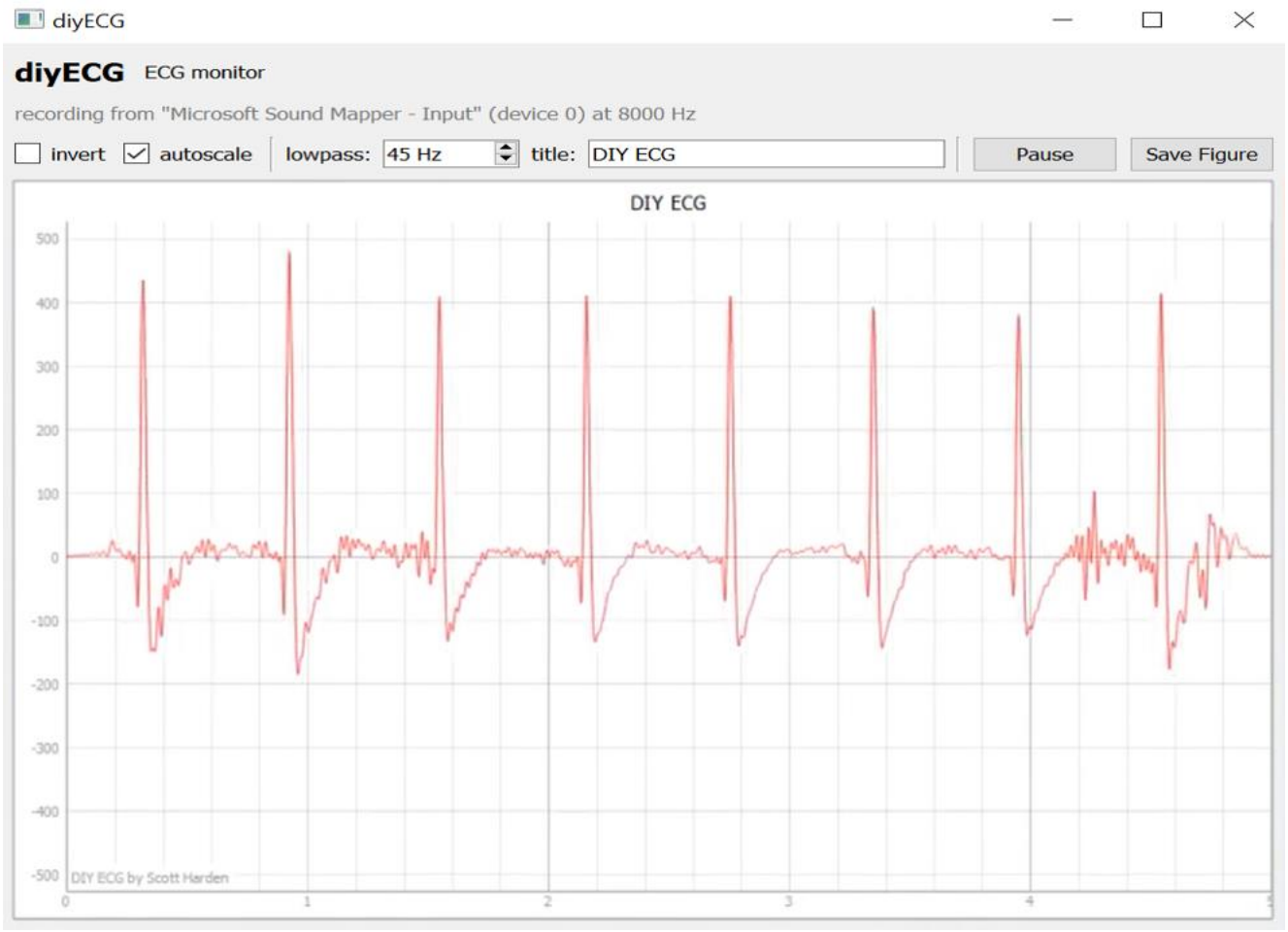

Figure 105 Live ECG after plugin microphone jack

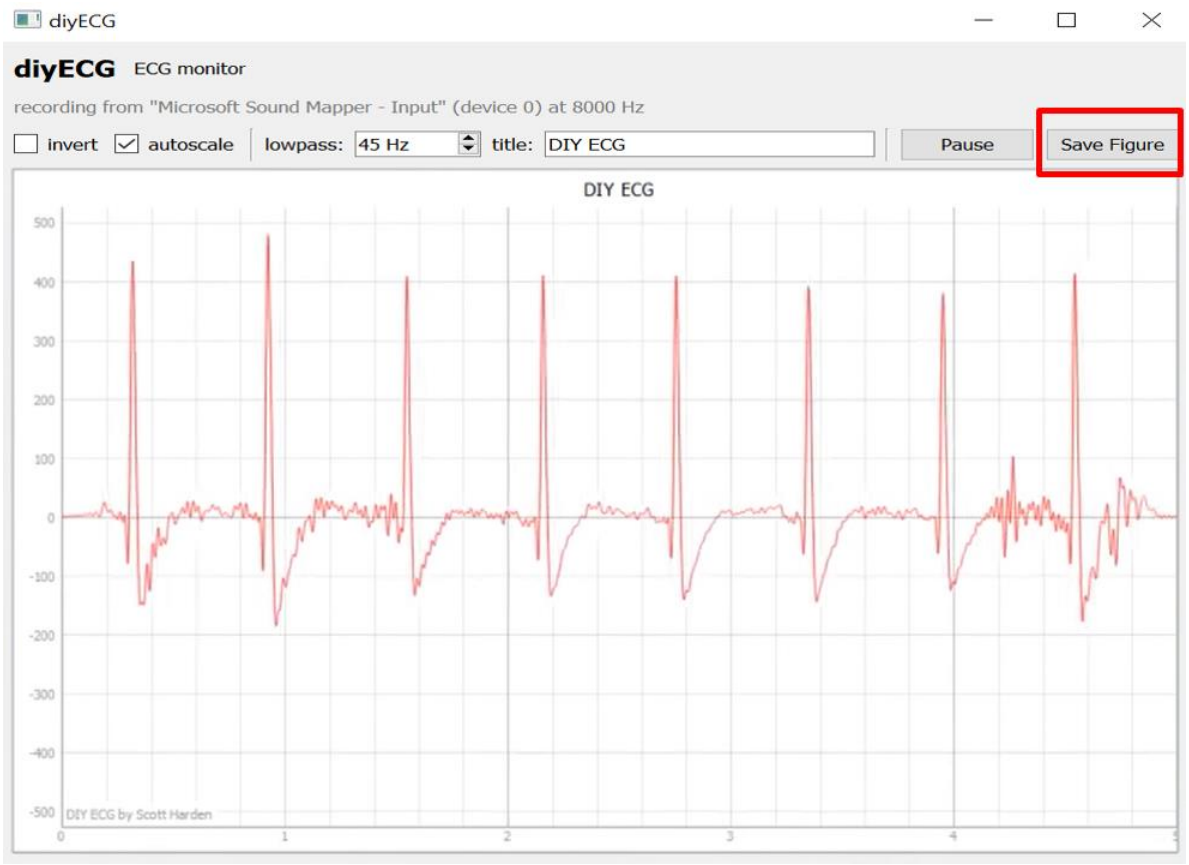

Figure 106 Saving the ECG data

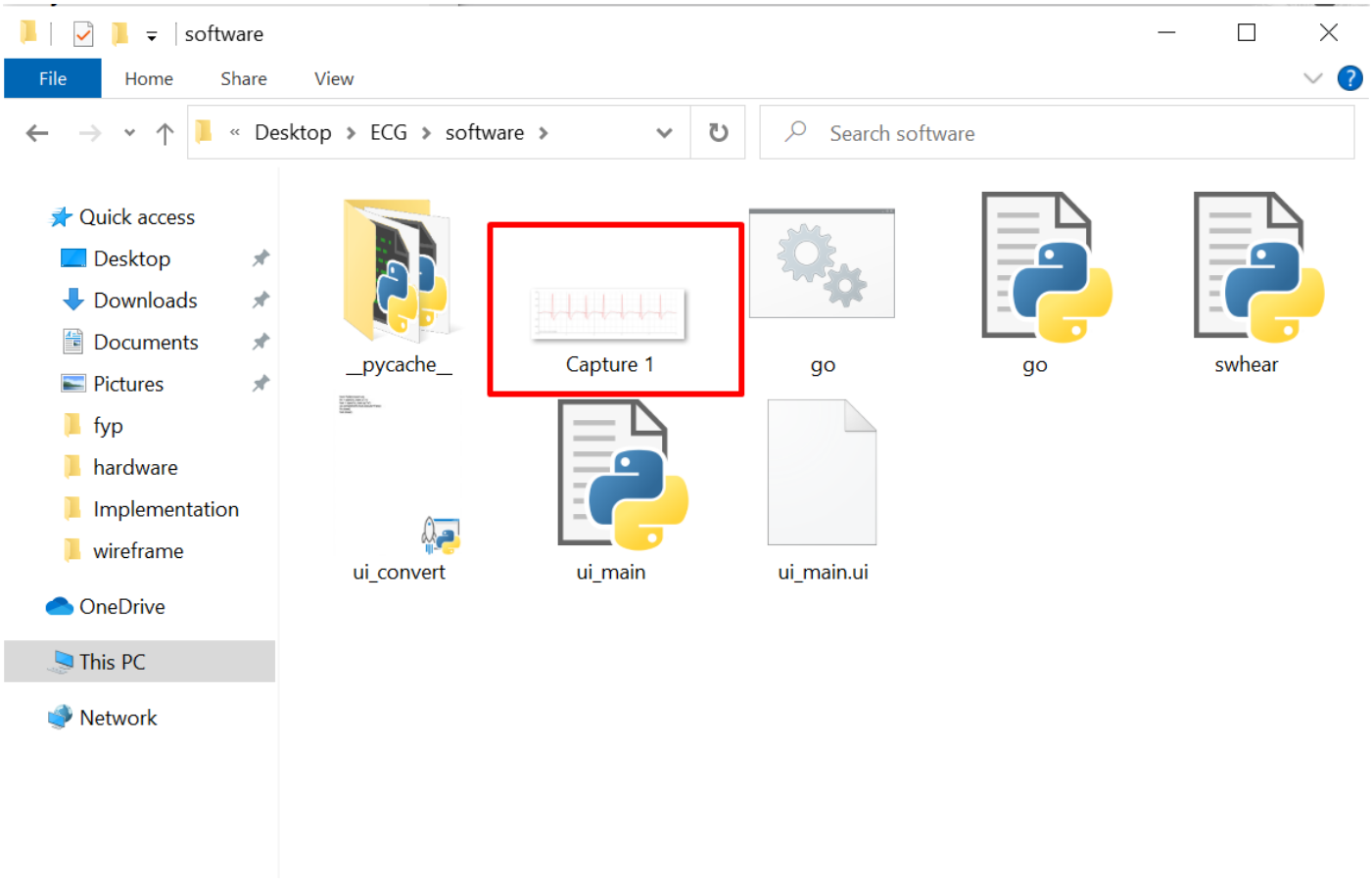

Figure 107 Saved ECG image

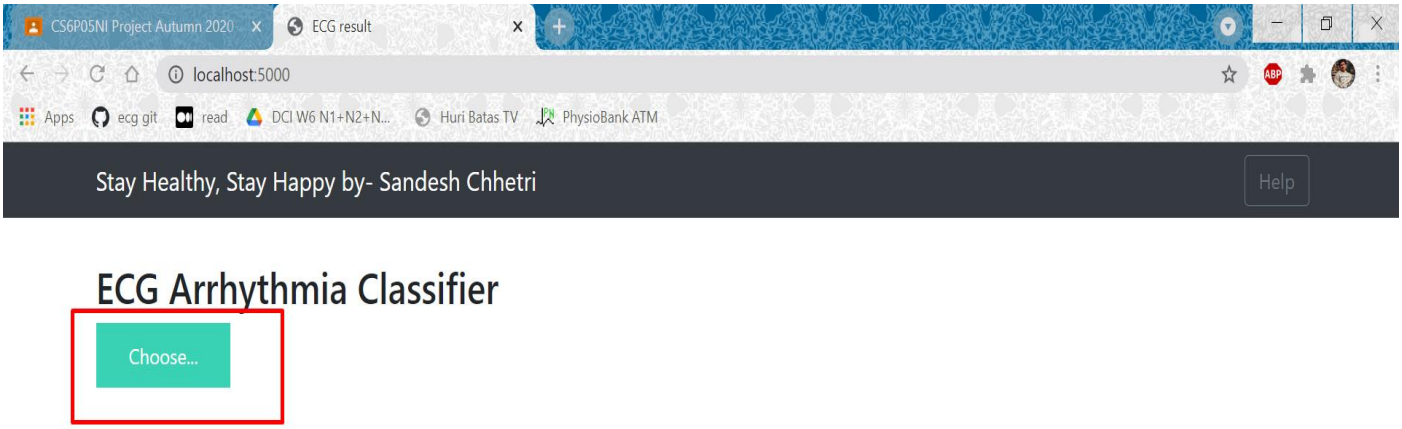

Figure 108 Uploading ECG data

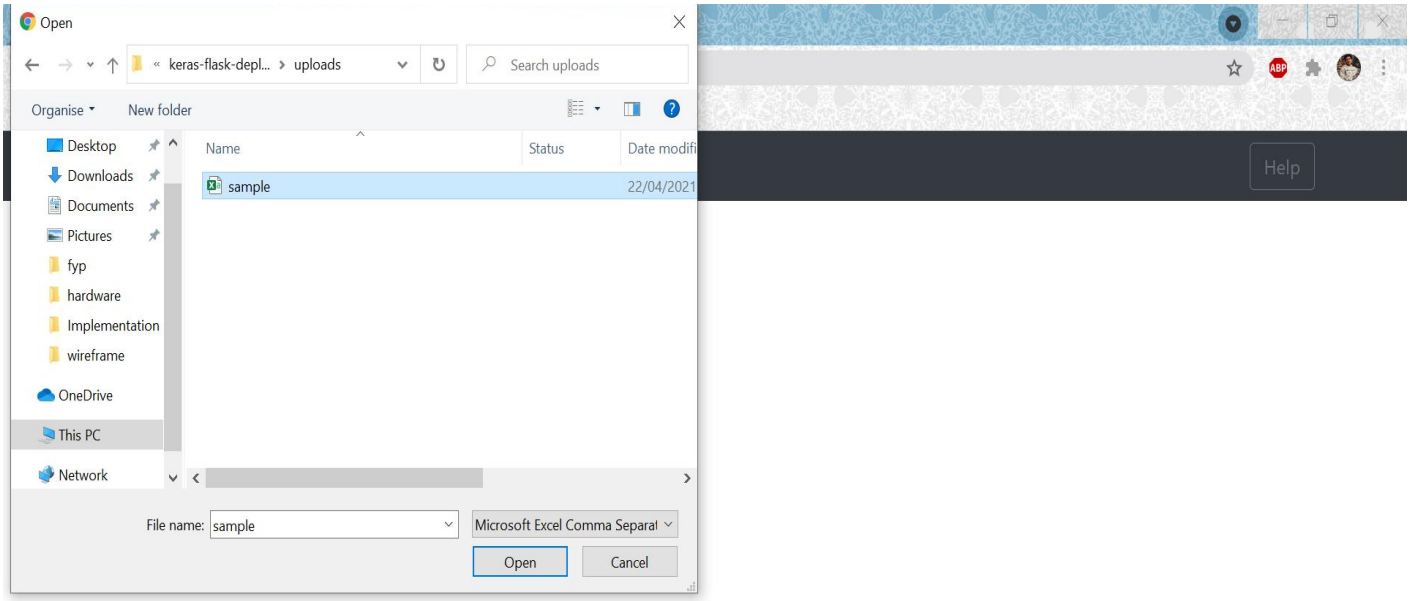

Figure 109 Uploading sample.csv data

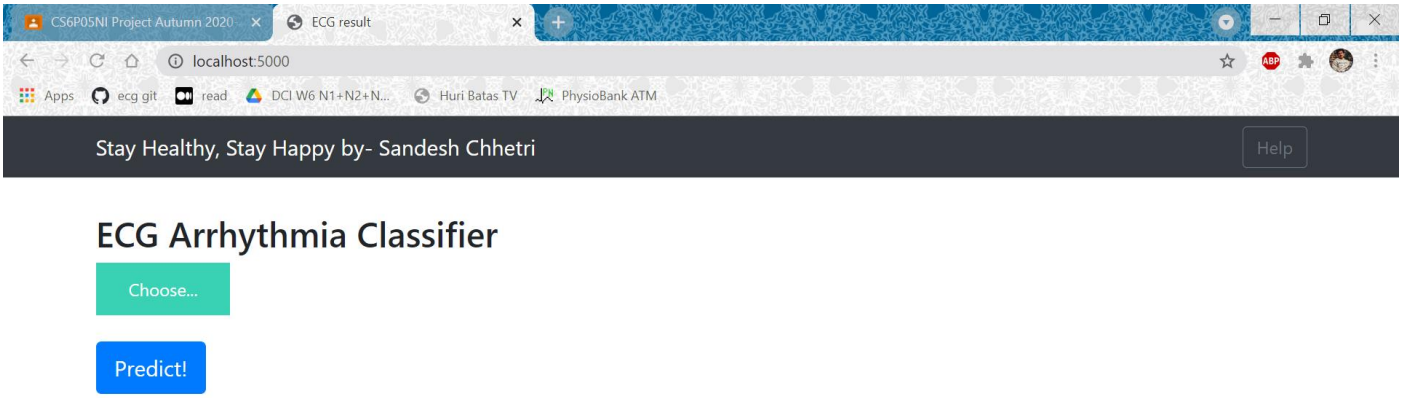

Figure 110 Clicking on predict result

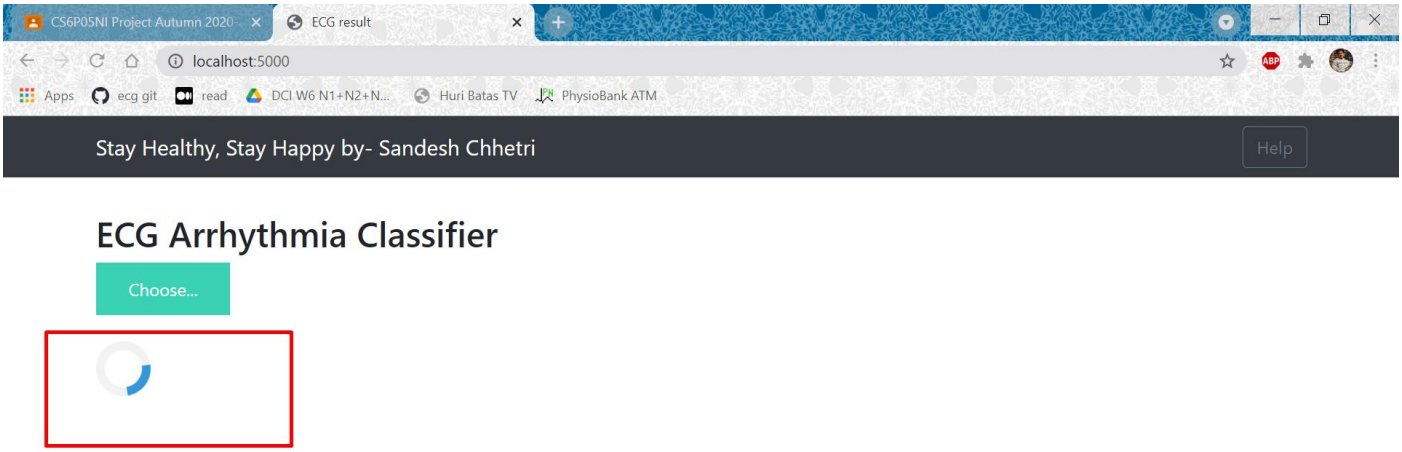

Figure 111 Predicting result from ECG data

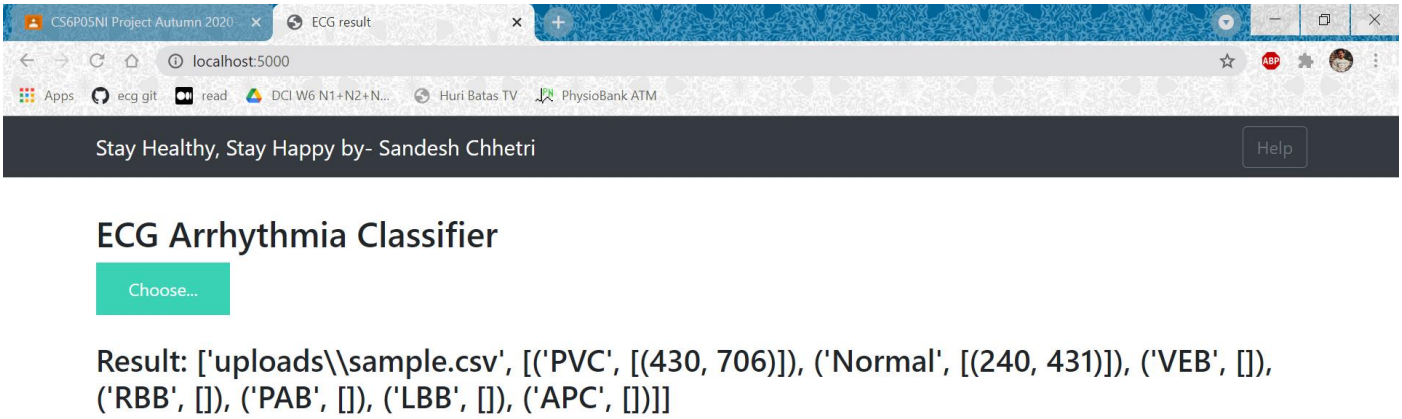

Figure 112 Result of ECG data

7.6 Appendix F: User Feedback

7.6.1 User feedback form

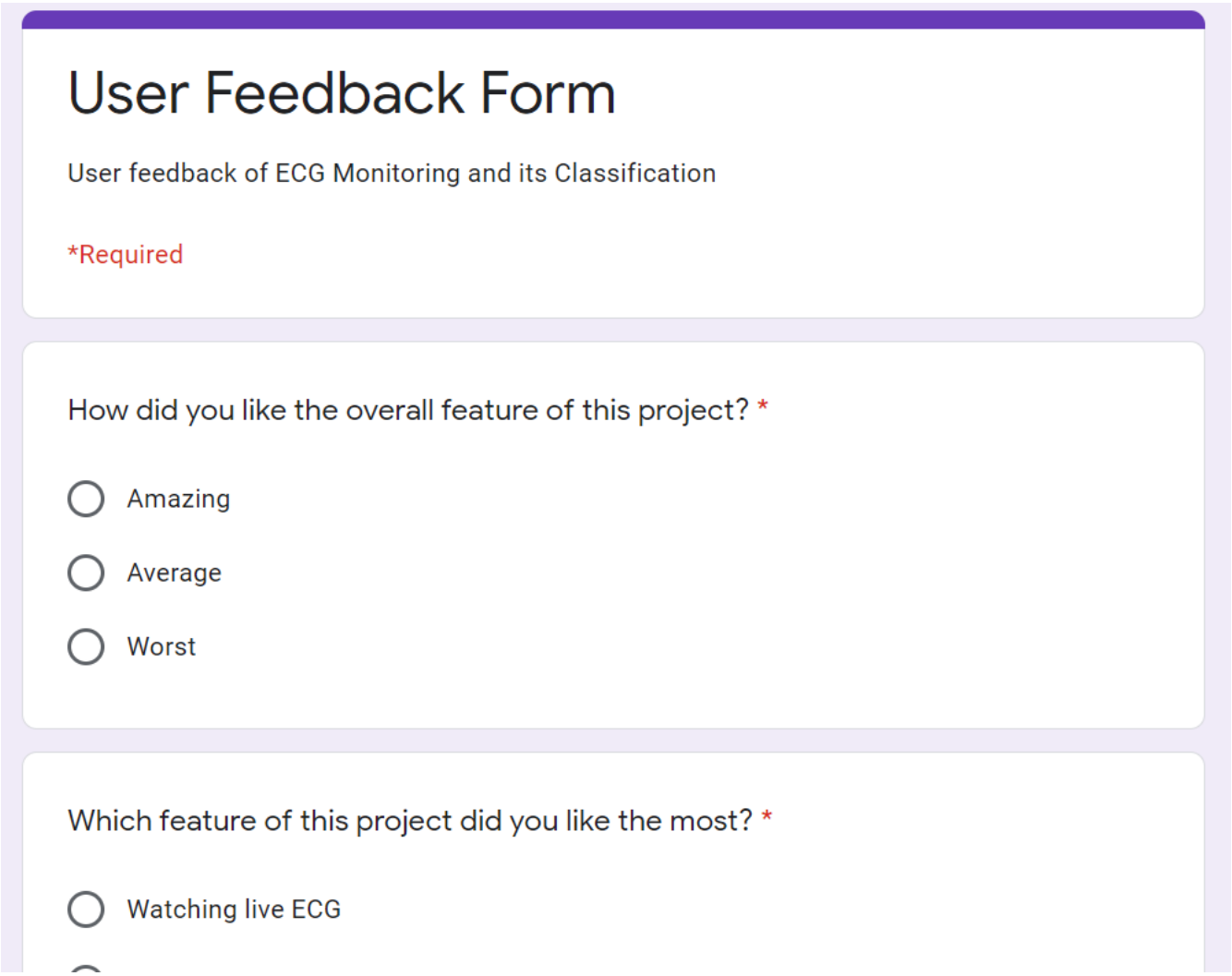

Figure 113 User feedback form 1

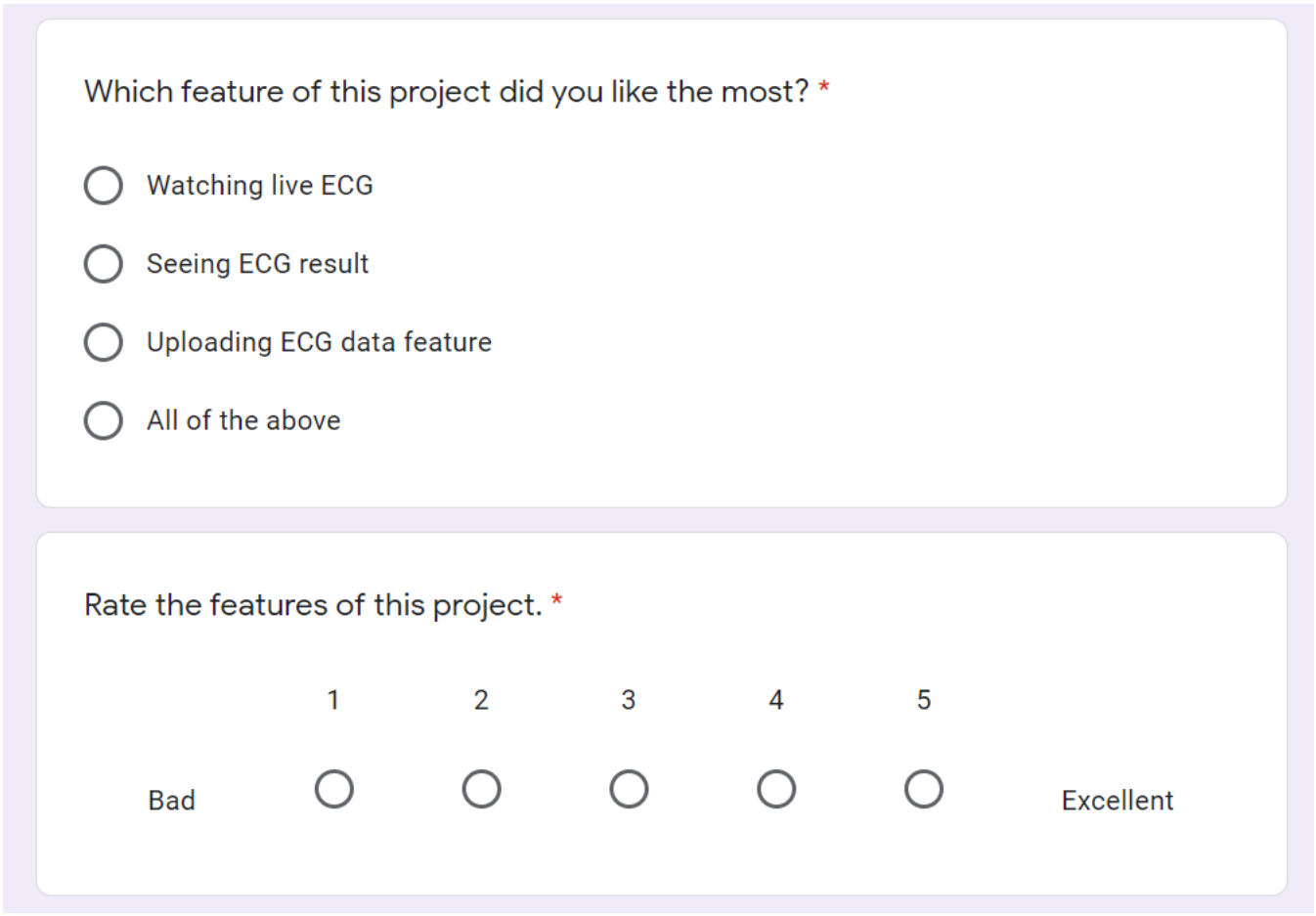

Figure 114 User feedback form 2

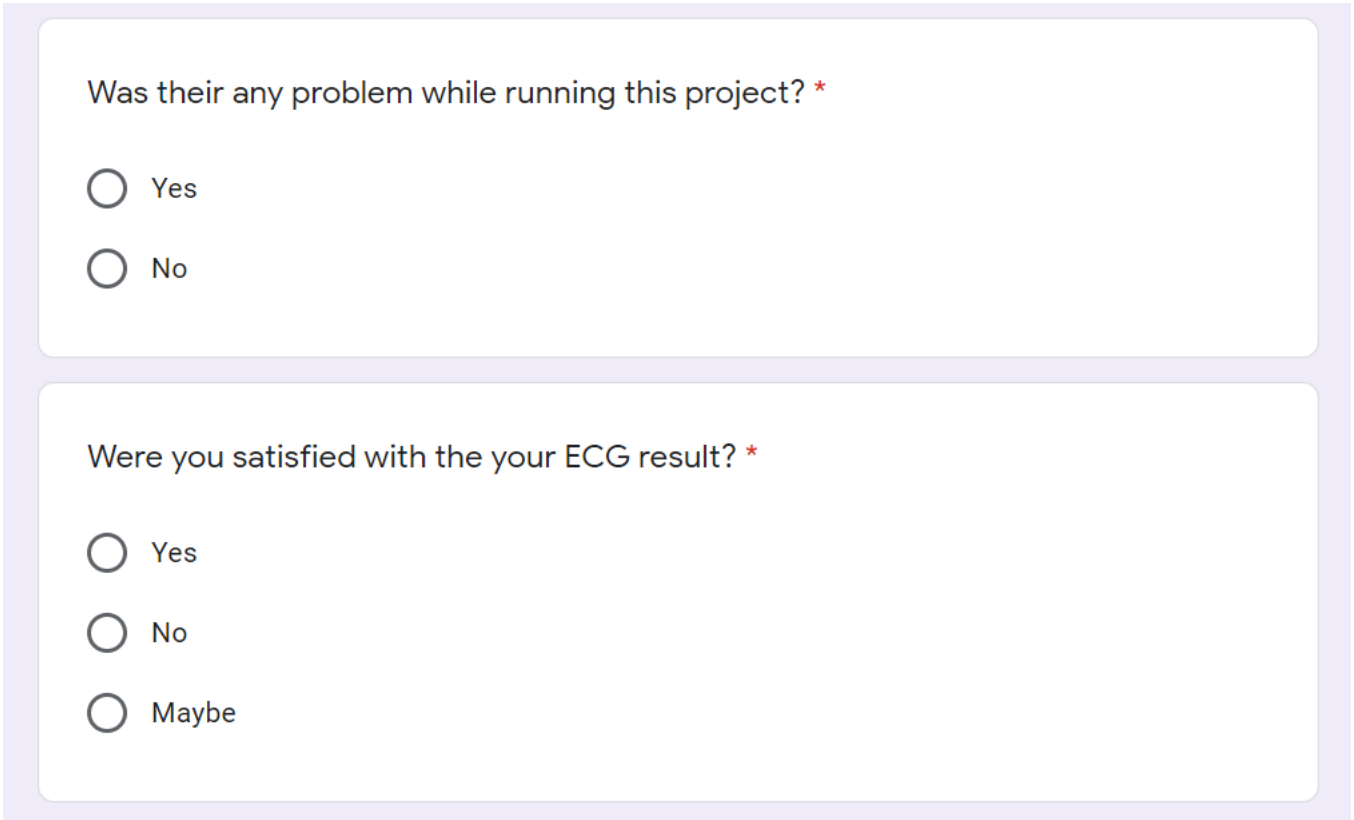

Figure 115 User feedback form 3

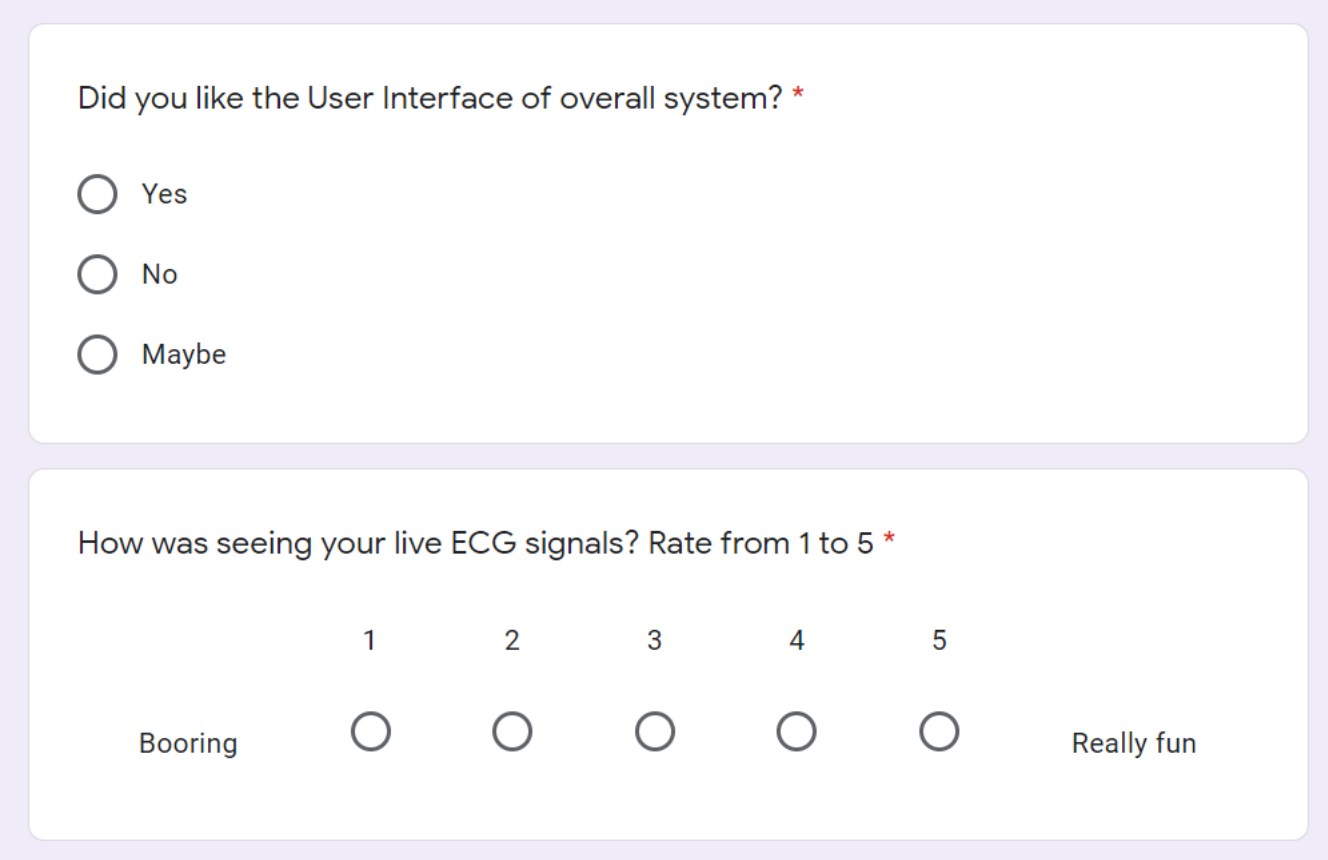

Figure 116 User feedback form 4

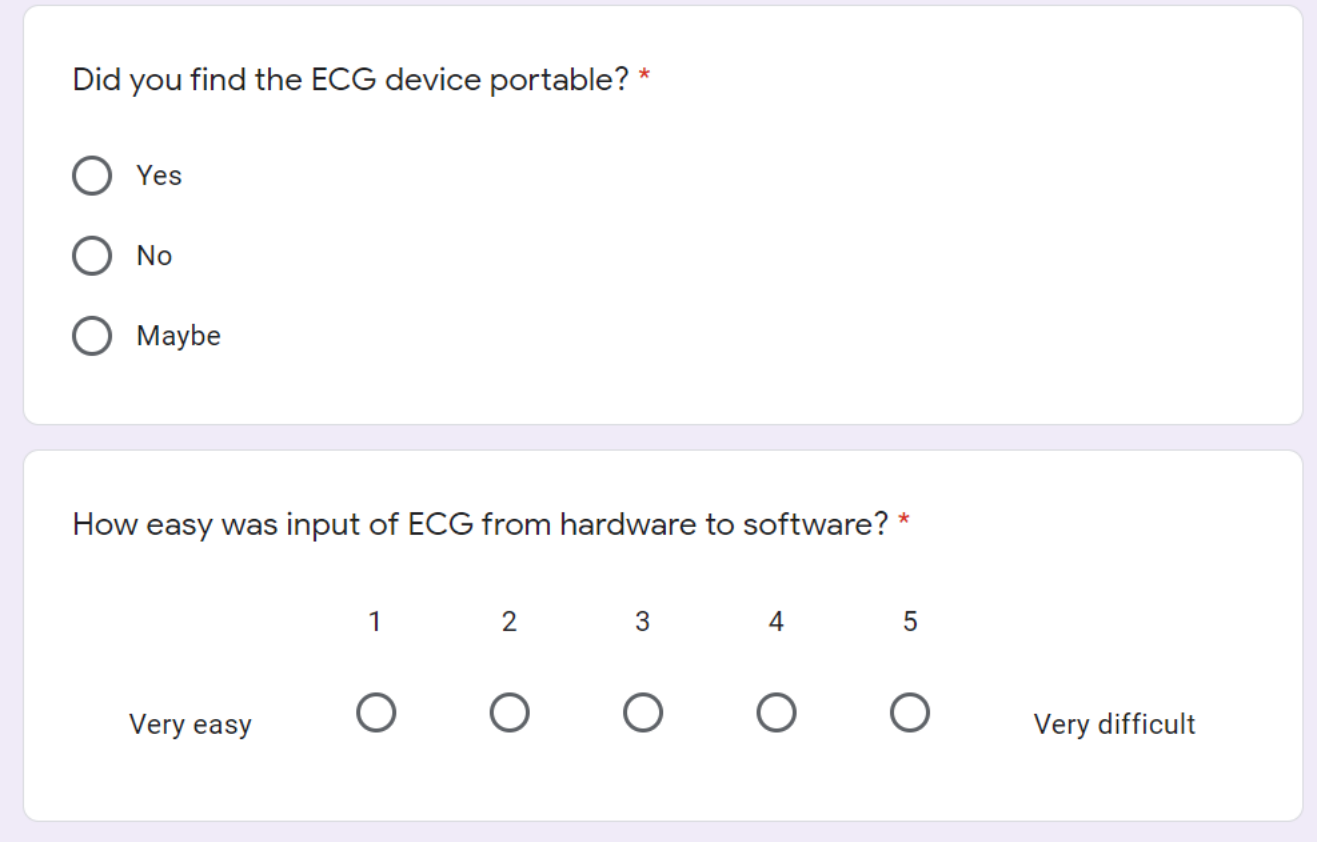

Figure 117 User feedback form 5

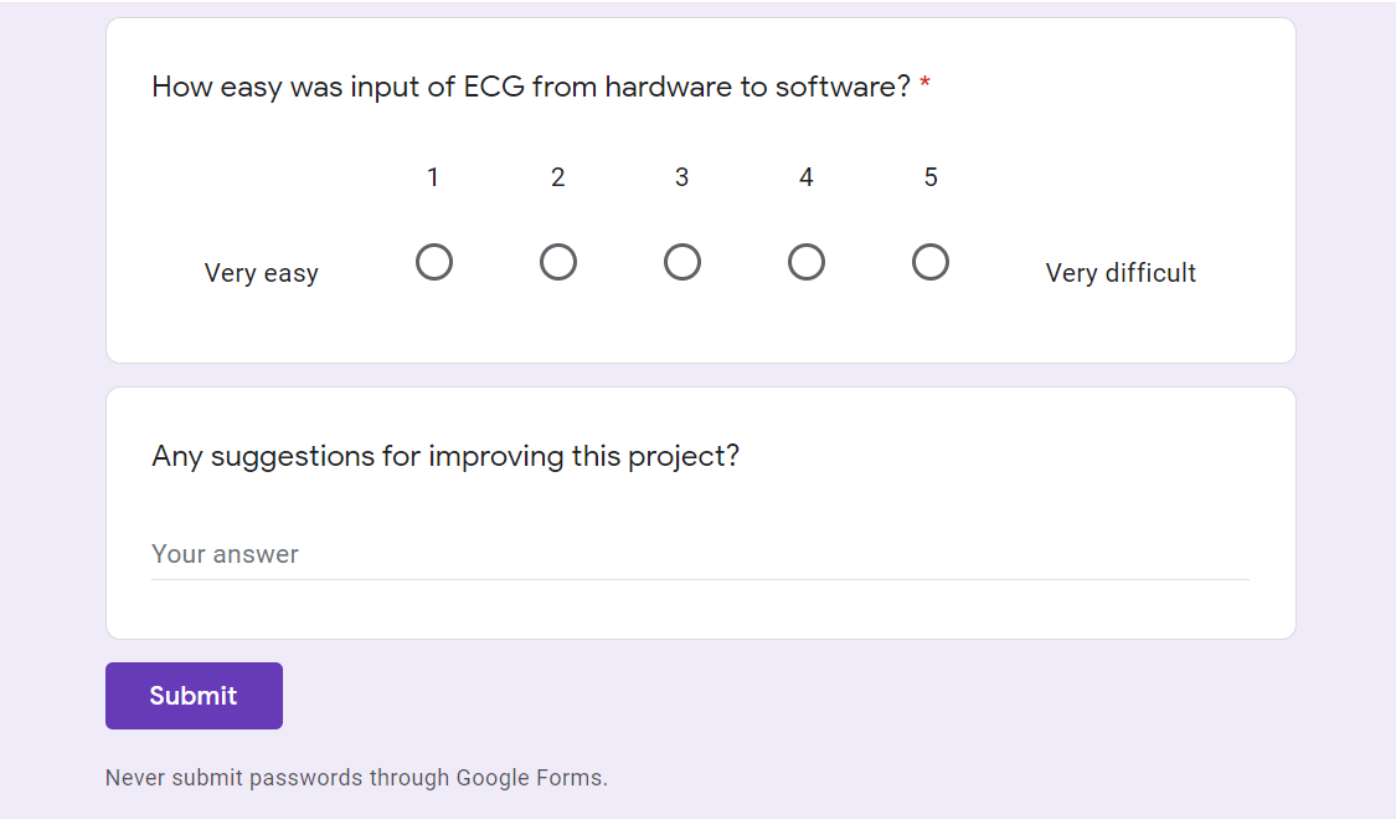

Figure 118 User feedback form 6

7.6.2 Sample of filled user feedback forms

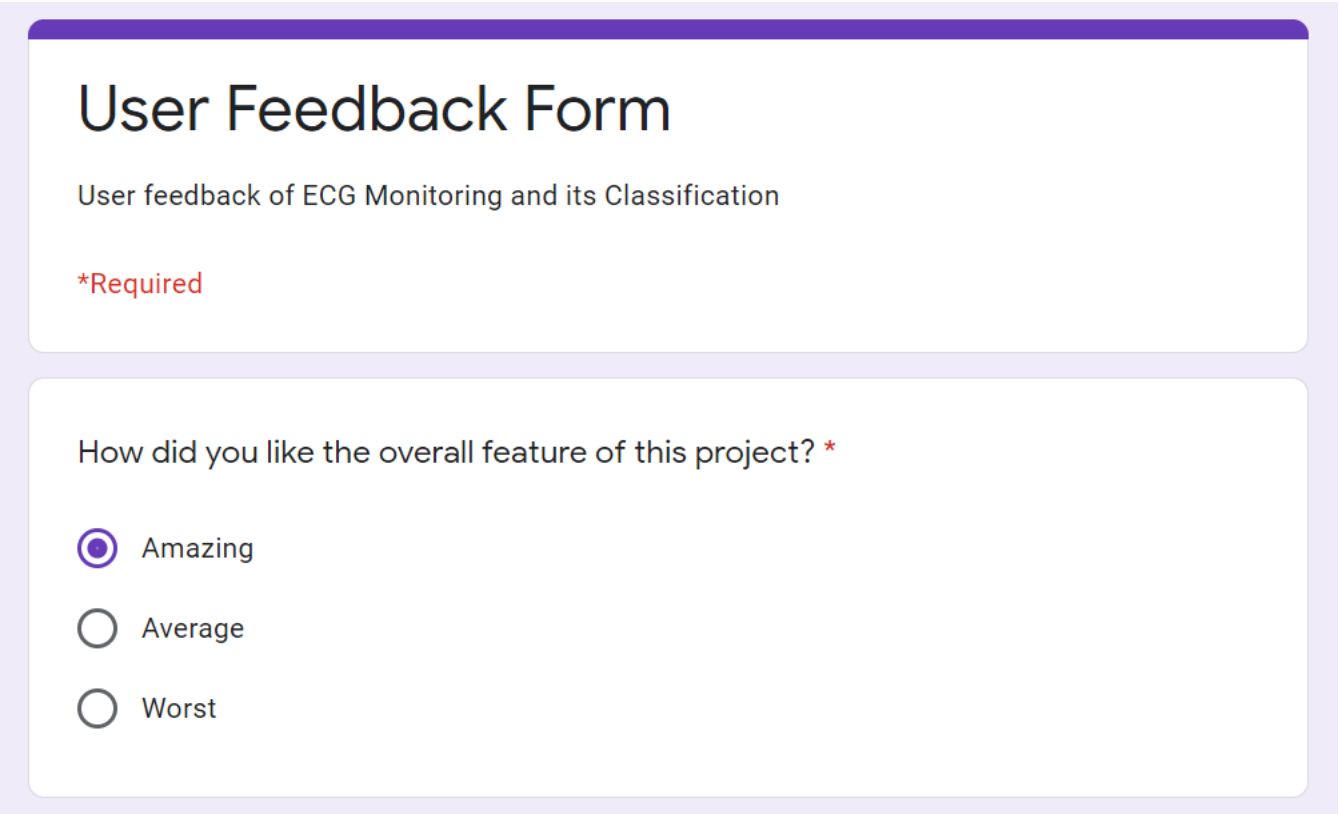

Figure 119 Filled user feedback form 1

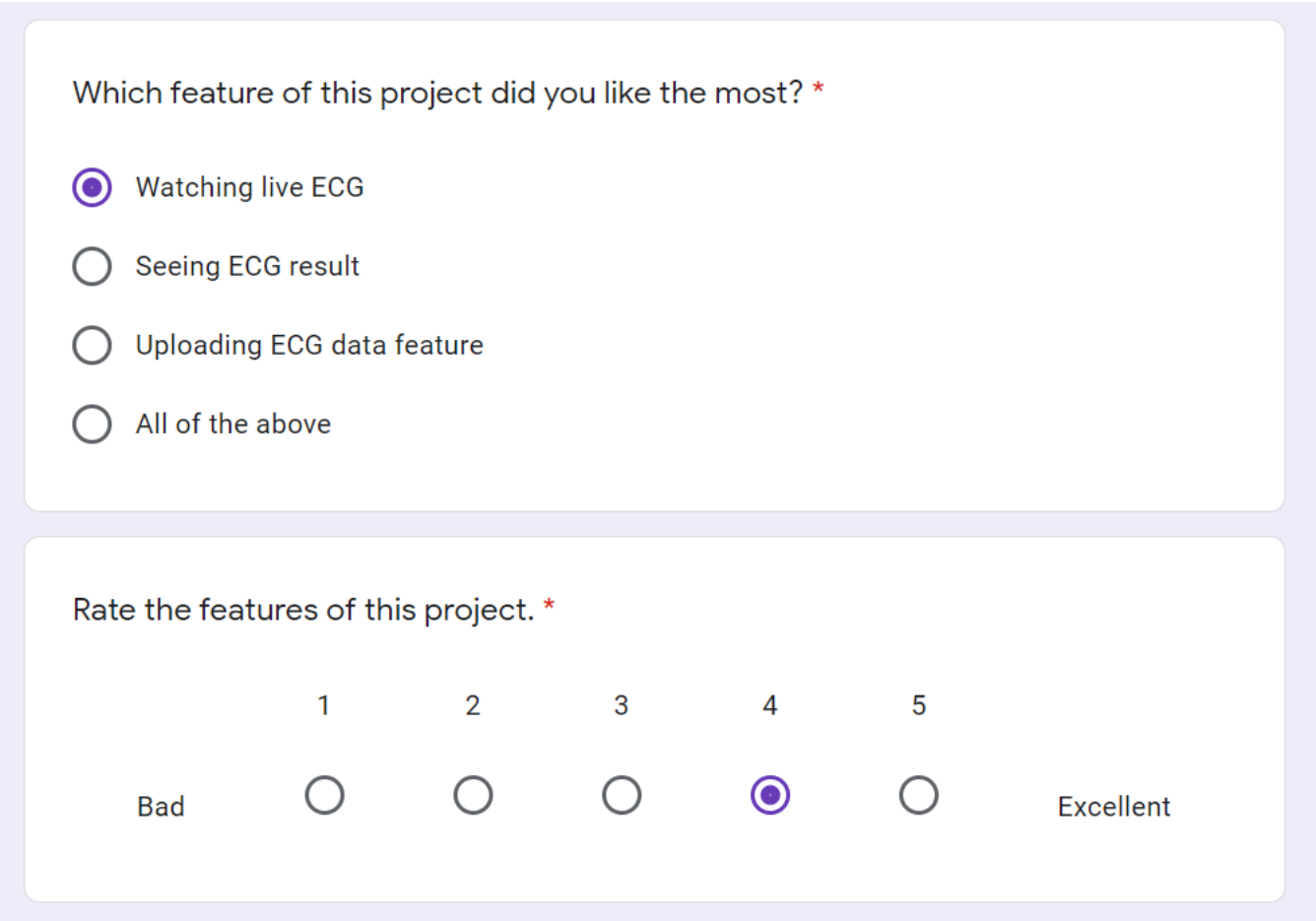

Figure 120 Filled user feedback form 2

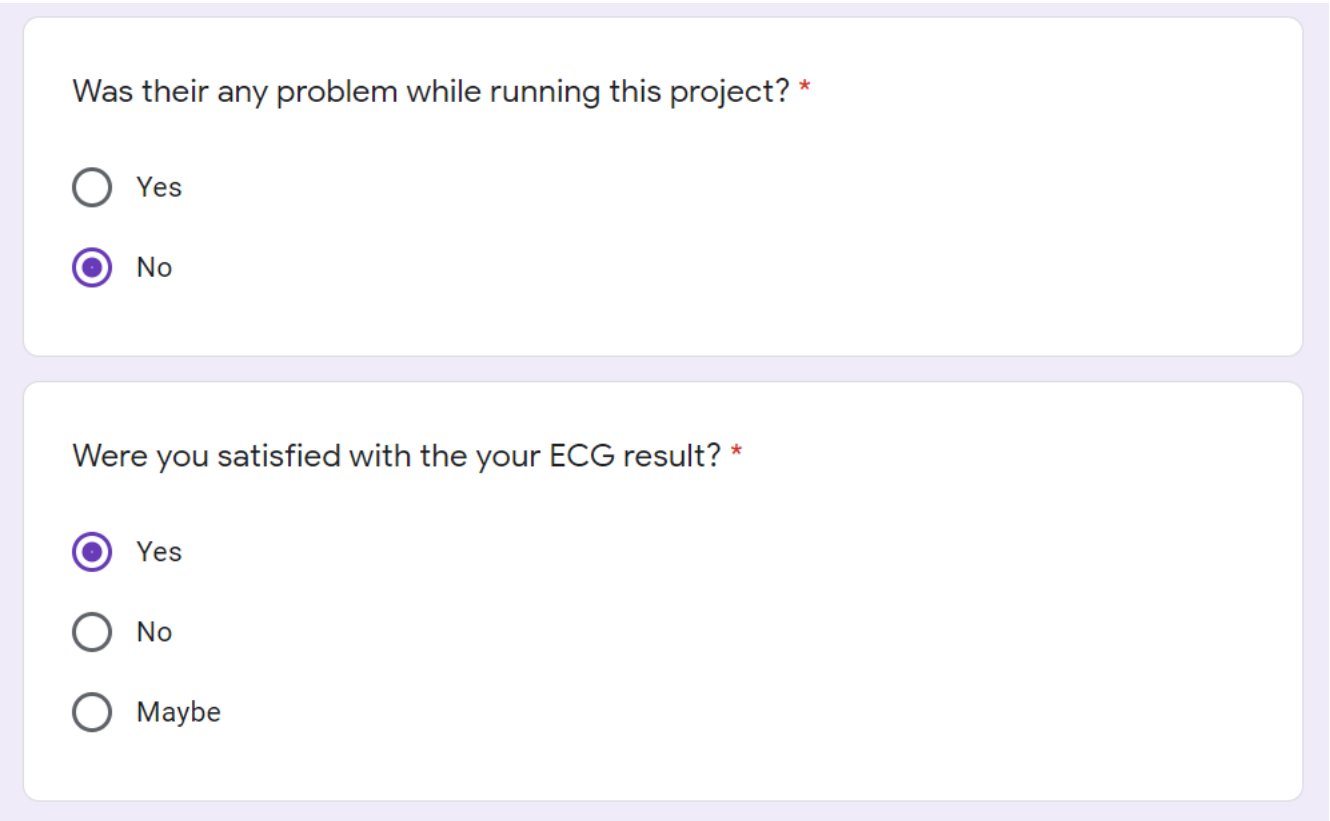

Figure 121 Filled user feedback form 3

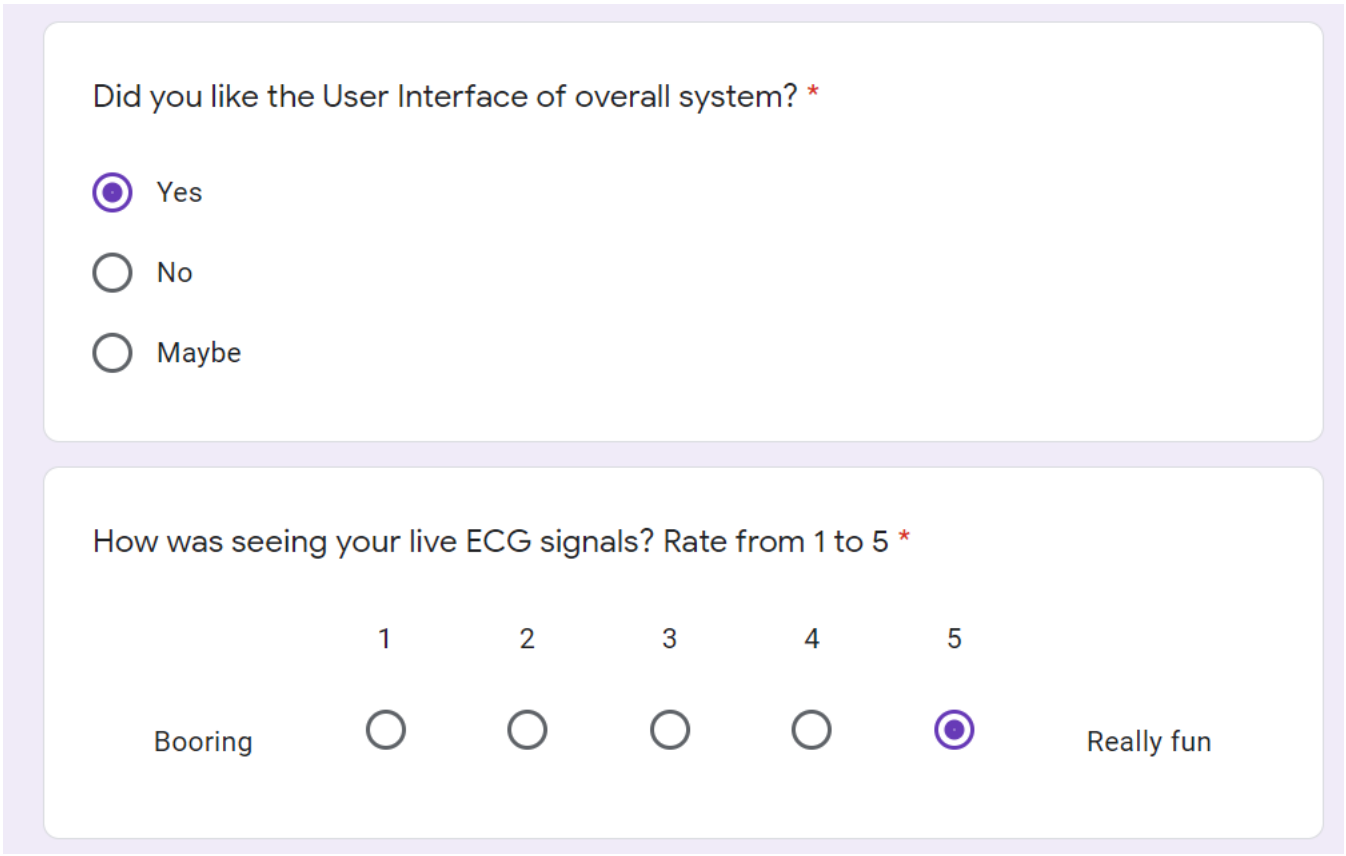

Figure 122 Filled user feedback form 4

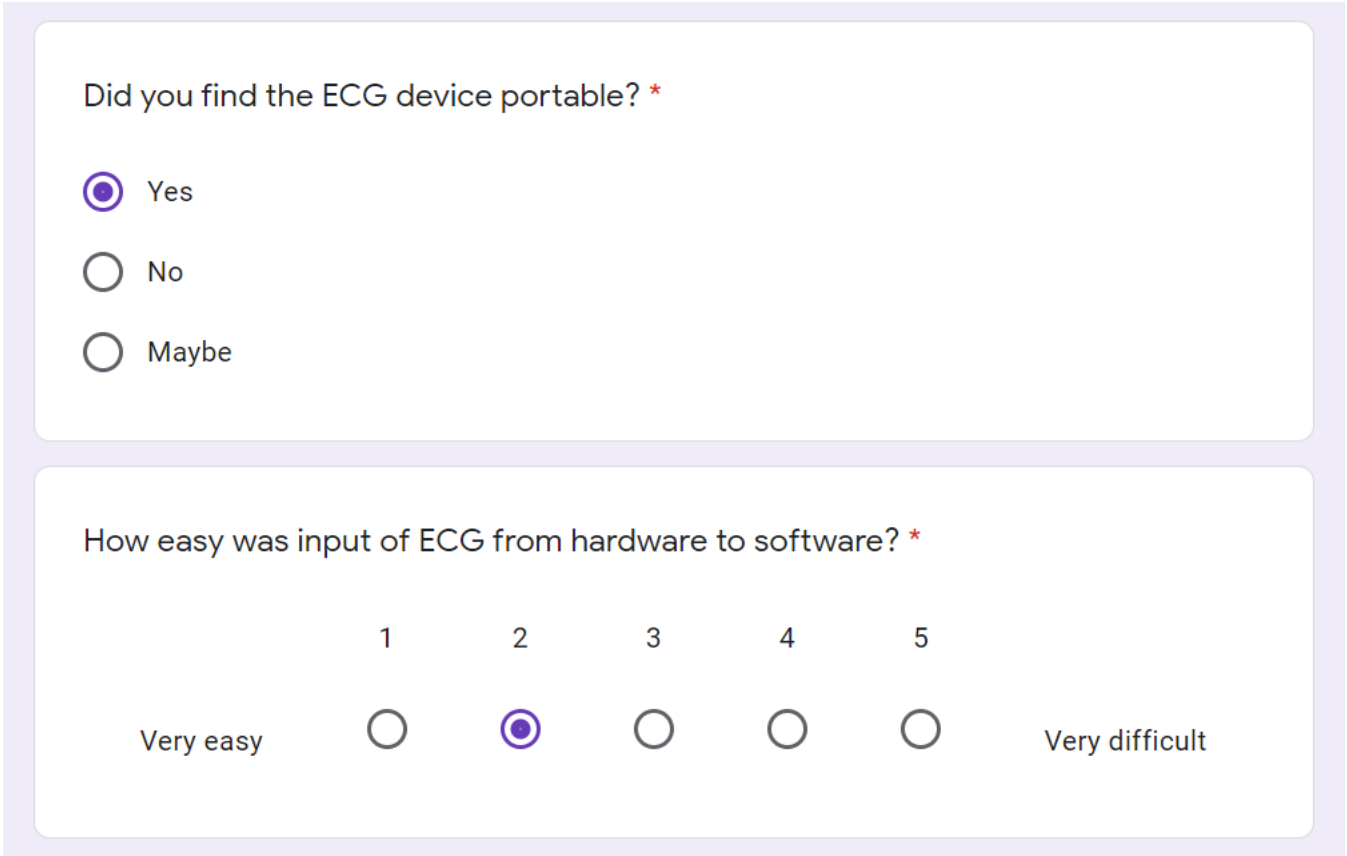

Figure 123 Filled user feedback form 5

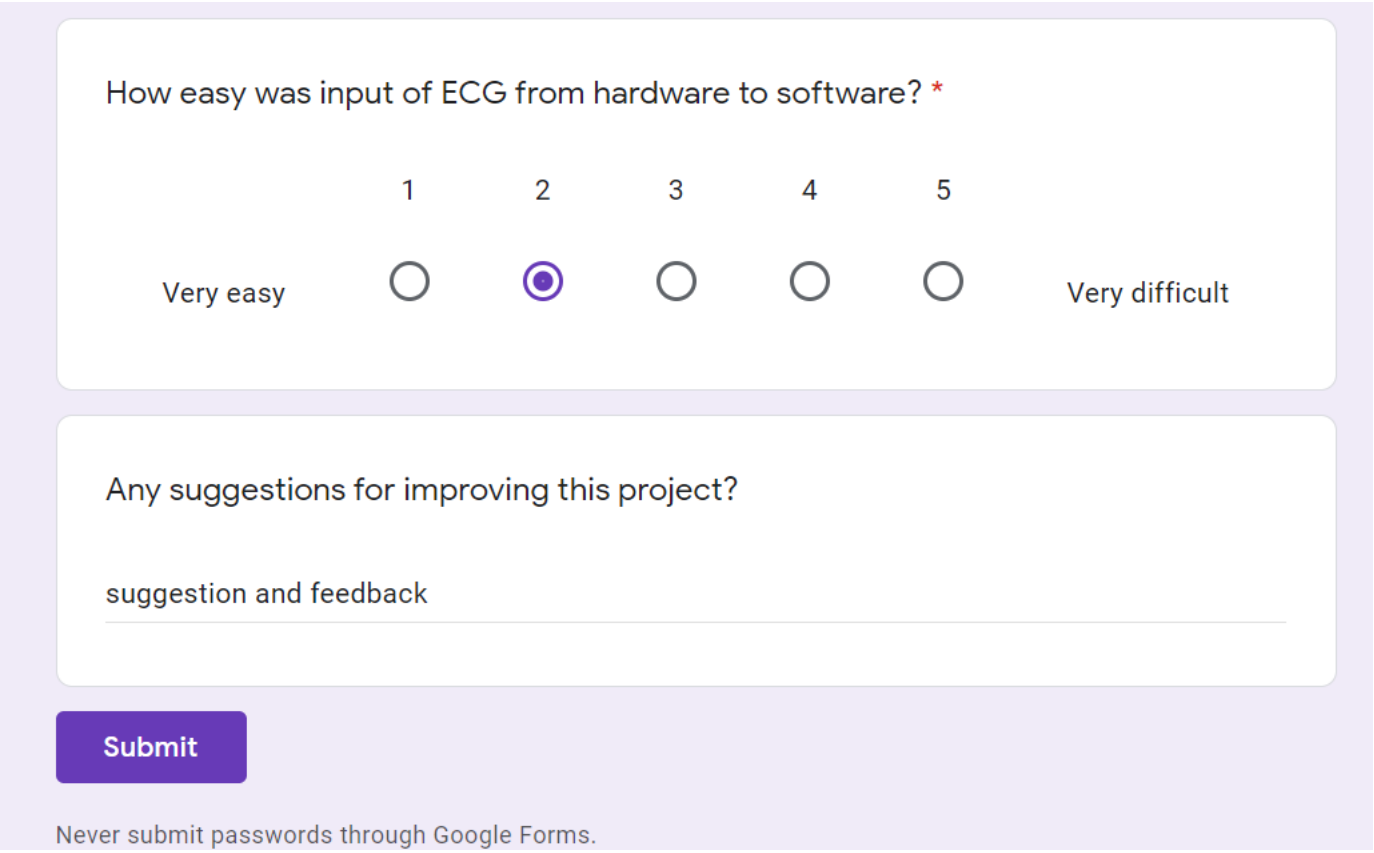

Figure 124 Filled user feedback form 9