UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Multimedia platform implementation for primary health centres in less developed countries.

A Degree Thesis

Submitted to the Faculty of the

Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona

Universitat Politècnica de Catalunya

by

Xavier Canyet I Quintana

In partial fulfilment

of the requirements for the degree in
SYSTEM'S TELECOMUNICATION ENGINEERING

Advisor: Eva Maria Vidal Lopez

Barcelona, June 2014

Abstract

This document attempt to detail the design, development and implementation of a multimedia platform for primary health centres in less developed countries. This project is a cooperation project due to it is a Doctors Without Borders or MSF(acronym for their french initials Médecins Sans Frontières) request.

The aim of this project is to set and implement a multimedia platform that allows MSF to obtain feedback at the same time users keep entertained.

MSF need to study and evaluate their service offered. So, they ask for a tool to generate, answer and present survey's results. In addition, we add the possibility to add some media (videos and games) to distract patient while they are waiting in waiting rooms.

Resum

Aquest document intenta detallar el disseny, el desenvolupament i la implementació d'una plataforma multimèdia per centres d'atenció primària de salut en països en vies de desenvolupament. Aquest projecte és un projecte de cooperació, ja que és una sol·licitud de Metges Sense Fronteres o MSF (acrònim de les sigles en francès de Médecins Sans Frontières).

L'objectiu del projecte és establir i implementar una plataforma multimèdia que permeti a MSF obtenir opinions i critiques al mateix temps que els usuaris s'entretenen.

MSF necessita estudiar i avaluar el servei que ofereixen. Per això, demanaven una eina per generar, respondre i presentar els resultats d'enquestes, A més a més, hem afegit la possibilitat d'afegir audiovisuals (vídeo i videojocs) per distreure els pacients que esperen a les sales d'espera.

Resumen

Este documento intenta detallar el diseño, el desarrollo y la implementación de una plataforma multimedia para centros de atención primaria de salud en países en vías de desarrollo. Este proyecto es un proyecto de cooperación, debido que es una solicitud de Médicos Sin Fronteras o MSF (acrónimo de las siglas en francés de Médecins Sans Frontières).

El objetivo de este proyecto es establecer e implementar una plataforma multimedia que permita a MSF obtener opiniones y críticas al mismo tiempo que los usuarios se entretienen.

MSF necesita estudiar y evaluar el servicio que ofrecen. Por eso, piden una herramienta para genera, responder y presentar los resultados de encuestas. Además, hemos añadido la posibilidad de añadir audiovisuales (vídeos y videojuegos) para distraer los pacientes que esperan en las salas de espera.

Acknowledgements

I would like to thank MSF and Eva Vidal for the given opportunity to collaborate in a cooperation project.

Special thanks to Dani Garcia who help me when I was at a standstill with HTML scripting problems.

Revision history and approval record

Revision	Date	Purpose
0	04/06/2018	Document creation
1	21/06/2018	Document revision

DOCUMENT DISTRIBUTION LIST

Name	e-mail
Xavier Canyet I Quintana	xaviercanyet@hotmail.com
Eva Vidal Lopez	eva.vidal@upc.edu

Written by:		Reviewed and approved by:	
Date	04/06/2018	Date	02/07/2018
Name	Xavier Canyet I Quintana	Name	Eva Vidal Lopez
Position	Project Author	Position	Project Supervisor

Table of contents

List of Figures

List of Tables:

1 Introduction

This is an ACOOP project. AUCOOP (in Catalan, stand for: Associació d'Universitaris per la COOPeració) is a UPC student association created to collaborate with less developed countries making projects that students can put into practise their tech skills learned with their studies.

This proposal belongs to a MSF and ETSETB (in Catalan, stand for Escola Tècnica Superior d'Enginyeria de Telecommunicacions de Bardcelona) projects pool led by Eva Vidal professor.

The undertaking is divided in two parts: technical and contextual part. The first one is explained in this document and the contextual part has been done by ESADE (in Catalan, stand for Escola Superior Administració i Direcció d'Empreses) students.

1.1 Background

MSF as any NGO (Non-Governmental Organization) offers a great deal of services to help the needy. In emergency and critical situation caused by wars and natural disaster, measuring beneficiary satisfaction is not such priority as other operations.

However, for MSF, the need to assess the quality of care provided by their field workers in their projects has become a key priority because they have detected the following problems:

- No evidence of the quality care offered.
- Lack of organised medical audit.
- Lack systematic outcomes assessment.
- Regulators have few formal monitoring systems.
- SAME not recorded (SAME stands for Severe Medical Adverse Event: defined as an injury resulting in prolonged hospitalization, disability or death, caused by healthcare management).
- Quality problems/misconduct/errors and their incidence are unknown in most of their missions.
- Limited structure and resources.

To resolve this, they have decided to create survey to ask their patients.

1.2 Statement of purpose

This project is a continuation of Marc Pons Serrano's project called Technologies for measuring beneficiary satisfaction.

The aim of that project was to create a server with a Raspberry Pi and offer a survey service. The project was presented and approved by MSF but, there was a problem with the proposed solution. The project's solution problem was that user needs internet connection to download app to be able to answer surveys created by MSF.

This project will resolve this by creating a web that will offer a survey service in addition to adding educational and entertainment resources.

1.3 Requirements and specification

MSF has specified the following requirements to this project:

1. Portable device
2. Autonomous device. Due to uninterrupted electricity supply is no guaranteed.
3. Energy efficient
4. No internet dependency
5. WIFI connection
6. Free open code software
7. Easy interface
8. Economic system

1.4 Methodology

This project has been developed in Raspberry pi using Raspbian and programming a web server to accomplish the previous requirement.

Rasbian is a linux version operating system. And it is free and open source.

WWW uses HTML JavaScript and PHP. So, any client can access the web app by any free own device's browser.

To have entertainment part in the web server includes videos and html games.

To provide original and proven successful games we include open free code html games search on Internet.

2 State of the art of the technology used or applied in this thesis:

2.1 Web Server definition

A client-server model is an architecture or structure where clients are provided resources and services by a server.

Clients are the end-user that sends the request to server. Clients can be any computerized device like computers, laptop, tablets, mobiles…

Server are central computer that serve all resources, such as files, songs, video… and other services.

A web server is a server that serve the resources and services to World Wide Web (www) and uses the HTTP protocol to send response and receive the client request.

WWW is a hypertext document allocation system. Hypertext documents are created using HTML.

2.2 Web Server Architecture

In dynamic web application architecture, we can identify the following:

- Client (browser).
- Web Server.
- Server-Side Scripting.
- Database

Architecturally, the design is usually viewed as having three layers.

- The Presentation Layer is responsible for maintaining the user interface.
- The Business Layer is where the business rules of the application are implemented.
- The data Layer is where the data is stored, that is the database.

2.3 Web Server functioning

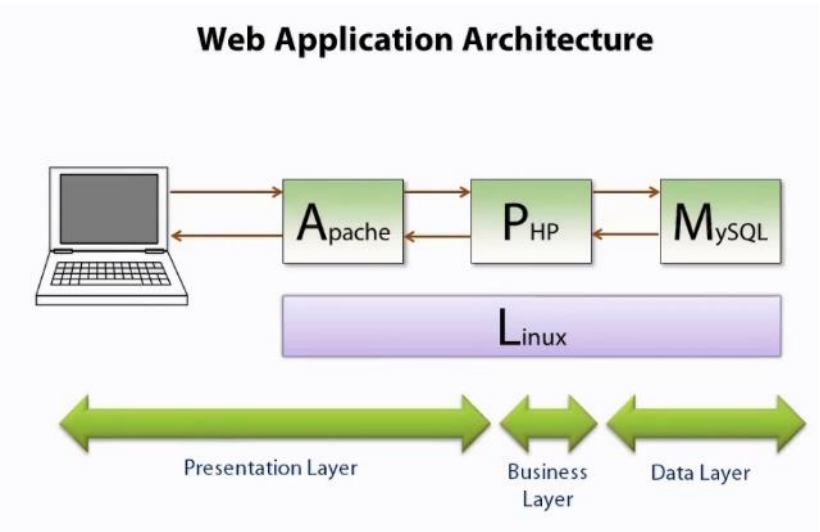


Figure 1 - Web Server Architecture

So, the Web Server basically operation is the following: a client with his device (PC, laptop, tablet, mobile…) search an URL in his browser. This is a request which is sent to a web server (Apache in this project), then the request is processed by server-side language (in this project PHP). In many cases, the script will consult backend database (in this project MySQL). The results of all of this is generate a HTML markup sent it back to the browser.

The browser is allowed to access and manipulate the objects on the page by JavaScript.

And the underlying operating system, in this project, is Linux.

2.4 HTML

Tim Berners-Lee in order to share research documents created Word Wilde Web an information space where documents (HTML) are identified by Uniform Resources Locators (URL).

HTML is a markup language used for structuring and presenting content to share across internet.

HTML stand for Hypertext Mark-up Language. Mark-up language means that is processed by client app (browser: Chrome, Internet Explorer, Firefox, Safari, Opera…) that is important point due to not all browsers support or have implemented all new elements.

After many years of evolution, the direction and development of HTML as formalized in the foundation of the World Wide Web consortium or W3C.This foundation creates the standards of HTML. HTML5 is a live standard that means new features developed are added to the spec at the time they are developed.

2.4.1 HTML structure

The research documents to share have text, data, image, charts… in them. So, HTML describe the web page semantic structure using tags.

There is always an opening tag <tag> and a closing </tag>. The structure is the following:

1. **<!DocType HTML>** standard of compliance. (HTML5, XHTML or past versions)

2. **<html>** wrap all html content we can include the language of the page.

3. **<head>** content all the not visually data to user but necessary for the client (browsers) such as:
    a. **<title></title>:** the tab's browser or window bar's browser name
    b. **<meta></meta>:** keywords and description for search engines and settings for browsers (content type, character set.)
    c. **<script></script>** all Java Script content.
    d. **<style></style>:** all the style body elements: position, size, colours…
    e. **<link></link>:** document related. (style sheets or JavaScript documents sometimes are too long and are separated of the document to be more readable)
    f. **<base></base>:** specify URL default target for all links on page.
4. **</head>** closing head tag

5. **<body>** opening body tag

Here in the body is where we include all the text, data, links, list, charts, tables, images, videos… of the whole page.

a. **<header>** jontly to <footer> signal the search engine the purpose for the markup inside this area.
    i. **<nav></nav>** logical container for a site navigation. (links to other documents or documents' section of the server)
b. **</header>**: closing header tag

c. **<main>** signal to the search engine that the most important information is here.
    i. **<article>** to give semantic meaning it is a way to group information
        1. **<section></section>** to give semantic meaning, to group related content logically.
        2. ** ** to group related content for visual purpose
        3. **<aside></aside>** to point information that it is important but not as necessary ad the rest of the information
    ii. **</article>** closing article tag
d. **</main>** closing main tag

e. **<footer>** opening main tag
    i. <nav></nav>
f. **</footer>** closing footer tag

6. **</boody>** closing body tag
7. **</html>** closing html tag

2.5 CSS

Cascading Style Sheet is a language to control the formatting, presentation, and overall look of a web page.

Every HTML element have style property that allows changing the layout (size, position, font colour, font source, background colour, …). But changing the style on each element individually is really mixing the structural UI information from the way it is displayed on a page.

For that reason, we use CSS to style different elements at the same time and to make reusable the style rules. We can reuse a defined style for more than one document, that way we don't have to copy all the style rules in each document, and it makes easier to edit.

By using a selector, we are going to give the instruction or rules (using name value pairs) for style all elements on the page with the name of selector.

Selector can be multiple (separated by comma ','). And we can use different properties of HTML elements such as: tag name (header,footer…), HTML element id, created label name, personalized class name.

Example:

```
selector1, selector2{
property: value1, [value2],…;
}
```

```
head, footer{
background-colour: red
}
```

2.6 JavaScript

JavaScript is a programming language that makes website interactive.

JavaScript is an **objected oriented** programming language. But uses prototypical inheritance instead class based inheritance.

JavaScript is **dynamic** means types can change and variables of certain types can change their types.

JavaScript is **interpreted language** that means it has the lack of compilation. But in most cases, on the browsers and in the server, JavaScript is 'just-in-time' compiled.

One of the different JavaScript form other programming language is that allow to get all the HTML with several ways:

getElementById,, getElementByTag

Beside that it has the same functions and operations as other programming language.

2.6.1 JQuery

JQuery is a JavaScript library for use client-side web development that allows create interactive websites that work across different browsers.

Before JQuery web development was difficult. It was much more difficult as the different browsers supported different things (diffenrent versions of Java, different versions of HTML spec).

The aim JQuery was to hide some of these differences so the developer could focus on adding functionalities not on cross browser.

JQuery is about querying and changing the DOM (Document Object Model) at the same time is very useful to event management.

2.7 PHP

Is a programming language. PHP stands for PHP Hypertext Pre-processor .

PHP its primary use is for creating websites. For this reason, it is usually embedded into hypertext meaning into HTML.

PHP is mainly focused on server-side scripting so allows to collect data, generate dynamic page content, send or receive cookies, get data from database server…

2.7.1 Connection to Database

PHP has two different ways to connect to a database:

1. PDO (PHP Data Objects)
2. MySQLi (MySQL Improve)

Both:

- Are Object-Orientated
- Supports Charsets
- Multiple Statements
- Server-Side Prepared Statements

The main two differences between them are:

- MySQLi can use procedural calls meanwhile PDO supports Client-Side prepared statemets
- MySQLi only connects to MySQL MaxDB, MariaDB meanwhile PDO connects to all DB supported in PHP

2.7.2 Retrieval data from database

Independent of the library used to connect to the database (PDO or MySQLi). In both cases, we execute a sql query by using a function/method (In PDO we use a connection's object method. Whereas in MySQLi, we call a function passing the connection by parameter.)

2.7.3 Pass Data from Client to Server

There are three different ways to pass the data dorm the client to the server:

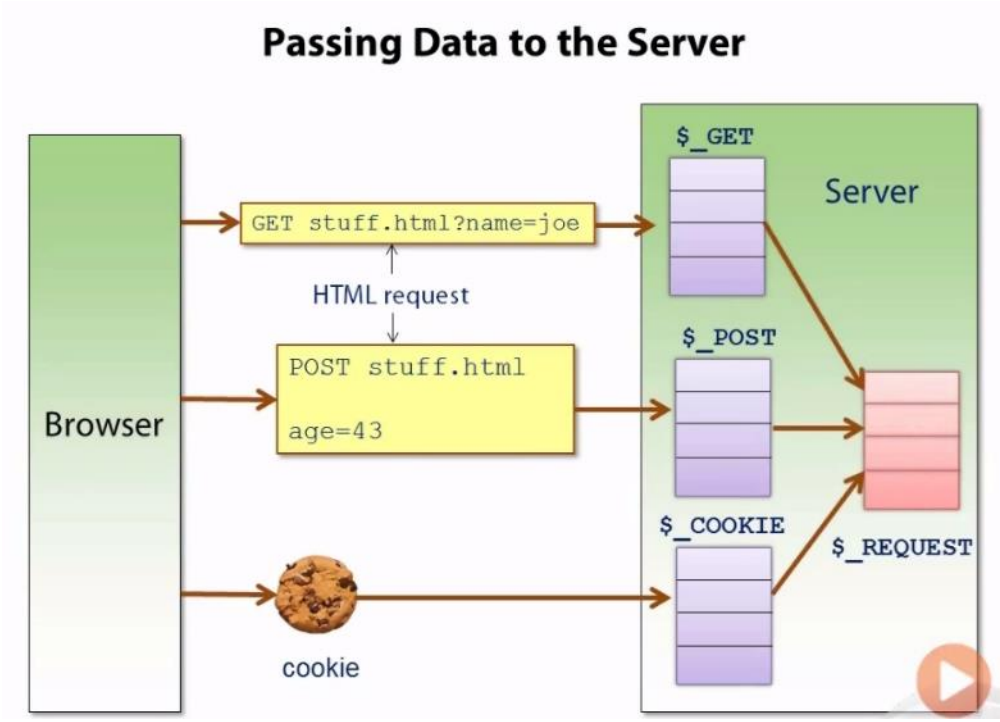


Figure 2 - Client to Server Data Transfering

2.7.3.1 GET method:

A set of name value pairs is passed in the URL. The information is collected on the server and is placed in '$_GET' associative array. (An associative array is an array whose index can be a string).

2.7.3.2 **POST method**:

The name and value pairs are passed in the message body. Information is placed in '$_POST' array.

2.7.3.3 COOKIES:

The values of any cookies sent by the browser are collected in the associative array called $_COOKIE

A cookie is a container of information (name value pair, domain, expire time ...) created on server and sent to browser for safe keeping. Browsers sends the cookies back to the server when it visits a related page on the site. Cookies are transmitted using fields in the HTTP request and response header.

There are two types of cookies:

- Session cookie
- Persistent cookie

Finally, the content of these three arrays is merge into a single array called $_REQUEST.

2.7.3.4 Sessions

To preserve data and access it across the entire website. In other words, for maintaining web page state information with a client, we must use sessions.

A session is a sequence of related interactions with a web application. It has specific goal. It ends when the browser is closed. It can be destroyed. Session uses cookies in background to work.

Eventually mention, PHP can be used for command line scripting and writing desktop scripting.

2.8 MySQL

A MySQL is the most used open source database (an organized collection of data). More to the point, MySQL is a relational database management system (RDBMS).

The name is composed by “My” name of cofounder daughter and SQL(Structured Query Language).

A relational database is collection data items organized as set of table and keys.

That allows tables to have relation that means:

- **1 to 1 relation**: 1 item from a table is directly connected to one unique item to another table. (e.g. one student has only one identification student number)
- **1 to many:** 1 item from a table is directly connected to two or more items to another table. (e.g. one specialization degree has many students)
- **Many to many:** many items from a table are directly connected to many items to another table. (e.g. one class have many students but one students take many classes.)

Relations allow database optimization. E.g. By deleting one entry, it deletes all related entries.

To work with tables, we use Query (request for information retrieval with database.)

The main query are the following (the correct syntax is not presented is a functional statement):

- **SELECT** column name 1, [column name 2], … **FROM** table name. [**WHERE** filter (e.g. column name == something)]**;**
- **INSERT INTO** table name (column name 1, [column name 2], …) **VALUES** (value 1, [value 2],…) **;**
- **UPDATE** table name **SET** column name 1 = value1, [column name 2 = value 2] [**WHERE** filter]**;**
- **DELETE FROM** table name [**WHERE** filter]**;**

Finally mention, that we can create users, create database, create tables, grant or revoke permission to users, and many things more with some SQL language on terminal or by using MySQL Workbench or phpMyAdmin tools.

3 Methodology / project development:

3.1 Introduction

In this section, the procedure done during the project will be detailed and explained

The project length it has been 5 months. The project has been realised during the 2018 spring semester from February to June. Nevertheless, the execution phase it takes only 3 months, due to, the first month was reserved for previous project completion and planning this project, and the last month was secured for documentation and test.

3.2 Setting Requirement and objective: Initial Set up

To accomplish the requirements mentioned before we take of advantage of previous studies and some decision made in the precedent project. In previous project, a Single Board Computer (SBC) market research was done ending that Raspberry Pi 3 was the most rentable due to its quality-cost relation-ship.

The first step was to install Raspbian a Linux version for Rasberry Pi. Once we have our Operating System installed, the second step was to configure Raspberry Pi as Access Point (AP) and configuring Raspberry Pi as a Web Server (LAMP) following the tutorials on official Raspberry Pi documentation.

After initial set up, the first decision token was to develop an own web page to solve the previous project solution problem satisfying the main project goal.

The previous project proposed a solution that require users to have internet connection access. In order to download a phone application to be able to answer surveys.

3.3 Research and Envisioning: Initial Demos

For this reason, the first part of the project was to study the different knowledge needed to develop a web:

- The client-server model and architecture.
- Databases design
- Server-side language
- Client-side language

All explained in the previous section: *2. State of Art*.

After a primary study and initials demos that allow user to store data in a database. The decision to create a web server was agreed with project supervisor in a first meeting.

To carry demo out the first step was to create a basic html form, a database with a simple table and write a few PHP lines to store data in the table.

Achieving this step is a major advance. This will allow to MSF to create surveys and users answer it.

3.4 Design: Create Database Model

Once proved that was possible to emulate a survey service. The following step was designing the database model.

Designing a database model involve:

- Decide how many tables will it take to store the information.
- Decide how tables are related.
- Decide table field
- Decide field types
- Decide the key fields

Next it is showed the database model:

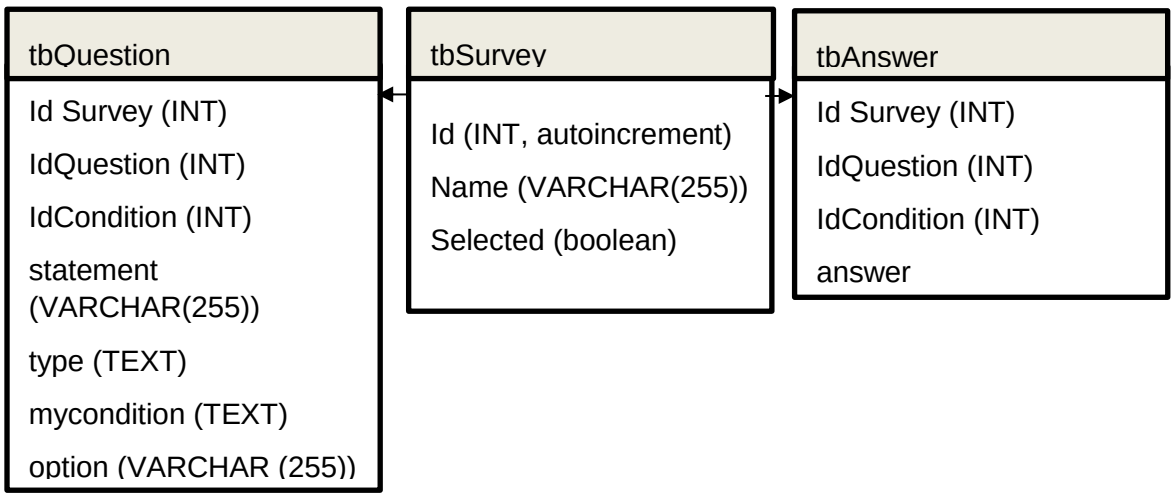


Figure 3 - Survey Database Model

3.5 Implementation: Process done

Web can be divide in two parts: *Survey Service* and *Education and Entrainment*.

3.5.1 Survey Service

Implement the first part implies to create a database with its tables and develop the html pages to emulate a real survey service. It means a form to store questions and a form to store the answers.

3.5.2 Education and Entertainment

The next goal was to illustrate to MSF all the option that can offer a personalised web. For this reason, new features were included in the web.

First, we include some health recommendation videos. Then, some html open code source games were included in the web server. Finally, the web was styled by including a home page, a navigation bar, a background…

3.5.3 Feedback and improvements

In the second meeting with project supervisor the work done was checked and it was agreed the next steps.

One of these steps was that question form improvement was mandatory. Due to survey creator don't have to know the back-end database structure.

Another step was to control the access to survey creation form.

And the last proposal was to categorise videos and group them by categories in the web visualization.

In the progress to accomplish these steps, a meeting with MSF was arranged and it was presented the unfinished project.

Figure 4 - MSF meeting

There was positive feedback from MSF workers who liked the work done, the idea and all the possibilities that offer a web server.

MSF workers requested that add a visit counter to the web server.

3.6 Problems

HTML 5 is a living standard that implies that when new features is defined it is included to the standard. Nevertheless, the client-side browser has the control of which features are included in it. It means that not all the browsers support all the features included in the standard.

This could be a major problem, because we have no knowledge about the supposed user's browsers. The supposed users may not have the last mobile models and the device used may not have the last browser's version. So, it is possible that they won't be able to use the last features included in HTML5 standard. Therefore, web page created must not include the newest features.

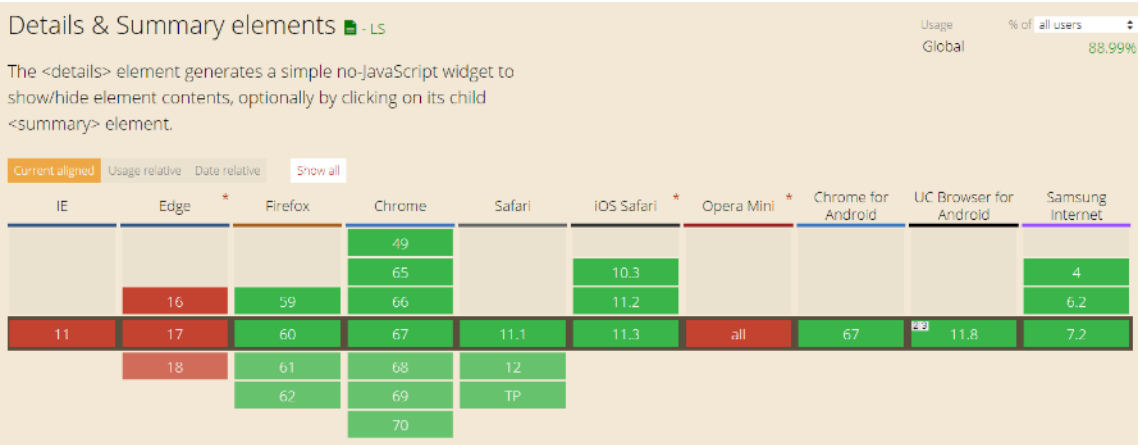


Figure 5 - Example: Details element support in different browser

To access to the web, user must connect to WIFI network enter it password, open his browser and search for Raspberry Pi's IP. To reduce complexity user can search Raspberry Pi hostname instead of Raspberry Pi's IP. There is a problem this does not work for all devices (it seems that works for Windows devices and iOS devices but not for Android devices).

4 Results

The project result is a first version of functionally web that can emulate a survey service and include media to distract patient while they are waiting in waiting rooms.

Next, it is display screenshots from the user interface of the web.

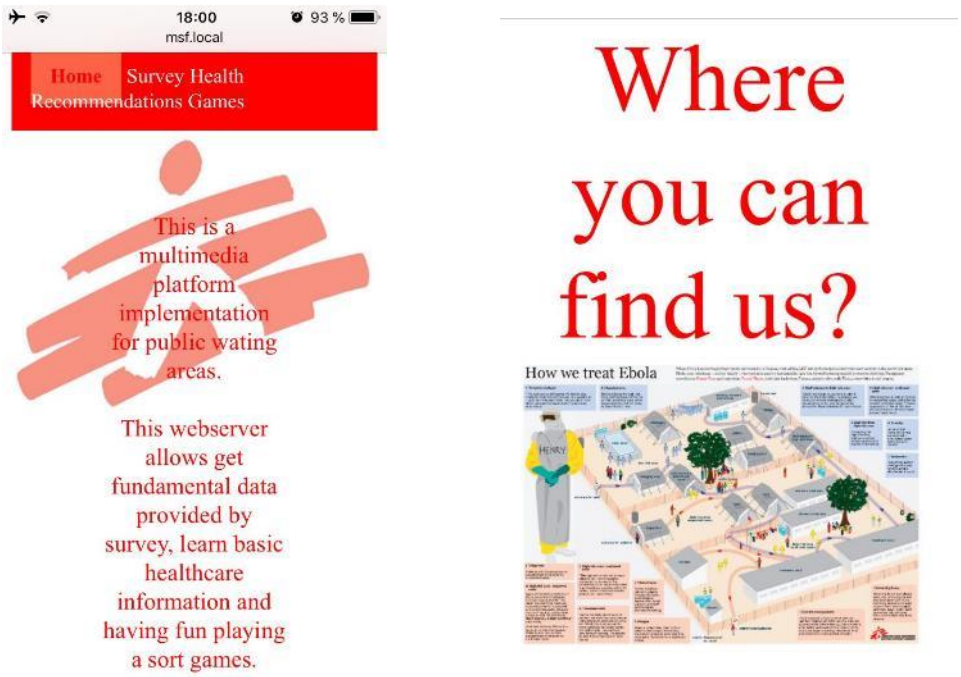


Figure 6 - Web Screenshot - Home page

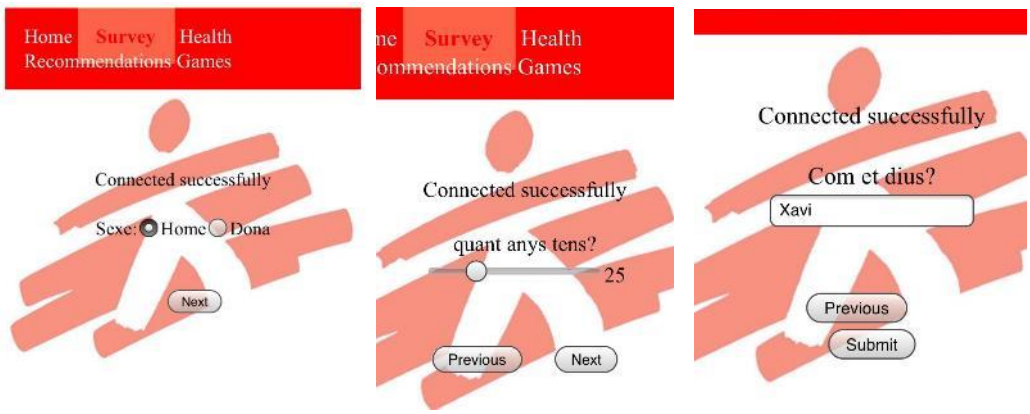


Figure 7- Web Screenshot - Survey Example

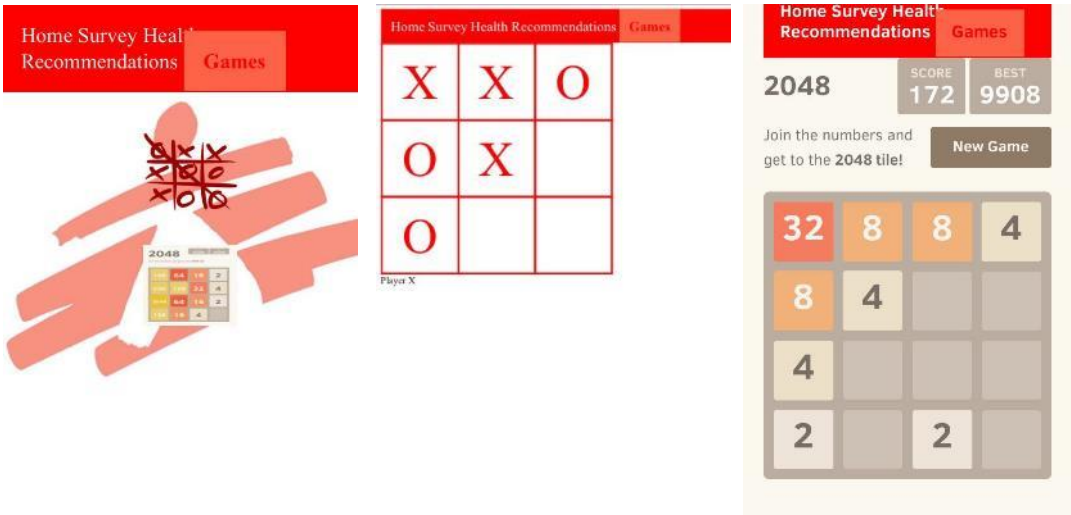


Figure 8 - Web Screenshot - Games

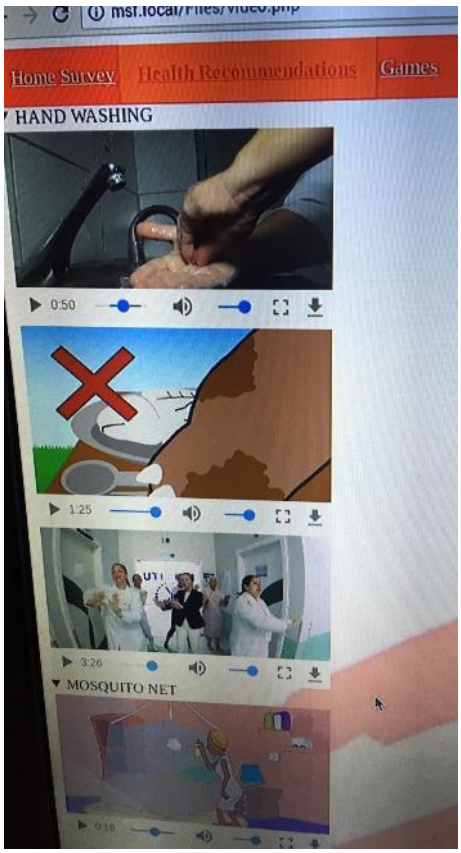


Figure 9 - Web Screenshot - Health Recommendation videos

5 Budget

In this section is presented the project budget.
As mentioned before this cooperation project to satisfy MSF request. Thus, this project has been developed to get minimum cost. To do that, we use free and open source code software.

Materials:

MATERIAL	UNITS	PRICE	TOTAL
Raspberry Pi 3	1	34,20 €	34,20 €
Raspberry Case	1	6,45 €	6,45 €
HDMI cable	1	25,45 €	25,45 €
64 GB SD card	1	3,75 €	3,75 €
TOTAL			69,85 €

Table 1 - Materials Budget

Supply:

OPTION A: Capable electricity power supply

MATERIAL	UNITS	PRICE	TOTAL
Raspberry Pi Universal Power Supply	1	12,00 €	12,00€
TOTAL			12,00 €

Table 2 - Supply Budget Option A

OPTION B: Electricity power supply impossible

MATERIAL	UNITS	PRICE	TOTAL
Power bank	1	40,06 €	40,06 €
Regulator	1	14,69 €	14,69 €
Photovotaic Panel	1	24,90 €	24,90 €
TOTAL			79,65 €

Table 3 - Supply Budget Option B

Staff:

ROLE	SALARY (€/h)	TIME (HOURS)	TOTAL
Junior Engineer	8	300	2.400,00€
TOTAL			2.400,00 €

Table 4 - Staff Buget

Breakdown hours:

Stage	TIME (HOURS)	PERCENTAGE
Setting Requirements & Objective	5	1,6 %
Research & Envisioning	30	10,0 %
Design	20	6,7 %
Implementation	180	60,0 %
Testing	20	6,7 %
Documentation	45	15,0 %
TOTAL	300	

Table 5 – Breakdown hours

TOTAL BUDET:

OPTION A:

Item	TOTAL
Materials	69,85 €
Power Supply	12,00 €
Staff	2400,00 €
TOTAL	2481,85 €

Table 6 - Total Budget Option A

OPTION B:

Item	TOTAL
Materials	69,85 €
Power Supply	79,65 €
Staff	2400,00 €
TOTAL	2549,50 €

Table 7 - Total Budget Option B

Many times, in cooperation project the human resources are not paid due to are voluntary services. For this reason, if we discount the Staff cost it is a economical project that can produce lot of social and educational benefits.

The possible benefits of this application are:

1. MSF can get a feedback of their service offered
2. Including health educational videos can be a good awareness campaign to prevent illness.
3. Perhaps games included can be enticement to make people go to primary medical centres to play, in countries where not always is possible to play video games.

6 Conclusions and future development:

This project has accomplished his main objective to offer a survey service with no internet connection dependencies. Patients will be able to access to Web Server by connecting to WIFI with their own devices and answer surveys, play health educational videos and play games to make the wait more bearable.

With this project I delved into WWW learning the basics of HTML, JS and PHP. I had discovered the problems of a living standard and browser compatibilities issues.

Besides that, my client server model knowledge has improved.

Finally, I learned to manage a RDB and how to structure data with a real application development.

This project will show the possibilities offered by portable web server to MSF. The Web application created may not be considered as Productive release. Instead it should be considered betaware or beta release.

This is project is a continuation of another project, but except the infrastructure, all software is new and nonprofessional because it has been done by myself. This means that have a lot room of improvement.

1. HTML, JS and PHP documents can be structured more clearly and then optimised.
2. Include validation rules to the form to insure valid user input data.
3. Make user able to change selected survey making that he can answer more than one survey.
4. Prompt a survey when user finish playing a video.
5. Totally personalized web. It means, to make admin user able to add photos, text, offering different styles and adding graph or whatever that can be desired from MSF doctors.
6. Add a presentation page where the MSF worker are presented to users.
    a. Create a table with doctor's name, specialisation, personal comment, and photo
    b. Present data in the web server using table element.
7. Queue management allowing the patient to know how many patients have before his visit

Making a web it will allow to include all the features you want. So, after the first pilot in a real MSF mission, with the feedback by doctors and workers all new requirements can be include in this server due to www versatility.

7 Bibliography:

[1] W3C Recommendation for HTML5, version 2 (HTML5.2), December 2017

[2] HTML 5 features [online] Available: https://platform.html5.org

[3] HTML 5 features browser support. [online] Available: https://caniuse.com

[4] MySQL manual [online] Availablehttps://www.mysql.com

[5] PHP manual [online] Available http://php.net

[6] Rasberry Pi configuration tutorials [online] Available: https://www.raspberrypi.org

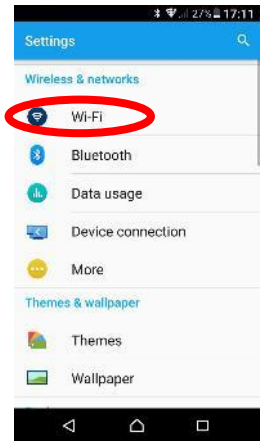
[7] HTML, JavaScript, PHP and MySQL demonstration [online] Available: https://www.w3schools.com

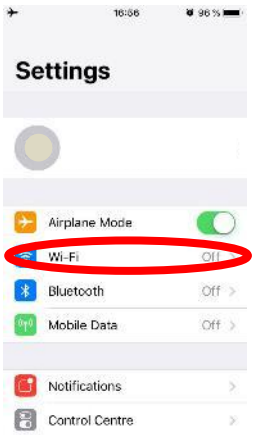
8 Appendices:

TUTORIAL TO CONNECT TO WEB SERVER

1. Open Settings on your phone

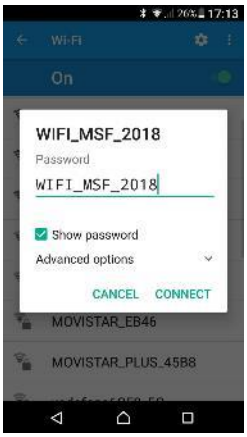
2. Enter on WIFI configuration





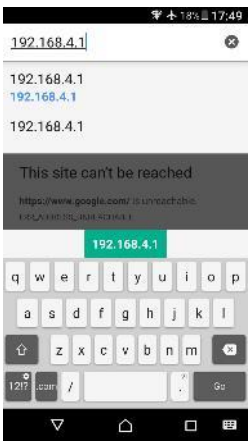
3. Select ***WIFI_MSF_2018*** network

4. Enter the next password: ***WIFI_MSF_2018*** enter pressing connect or Join



5. Open your browser

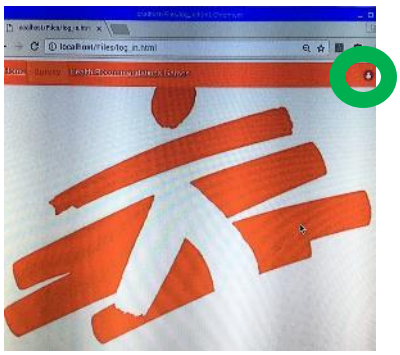
6. Search **198.162.4.1** or msf.local on your browser



7. Then you have access to web

TUTORIAL TO CREATE A SURVEY

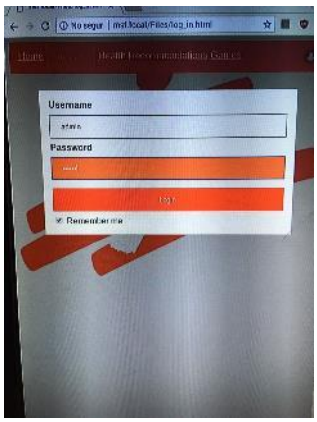
1. Once accessed to web, you must log in with administrator user. Press the icon on the right top corner of the page.



2. Register to the pop up form with the following credentials:

Username: admin

Password: admin

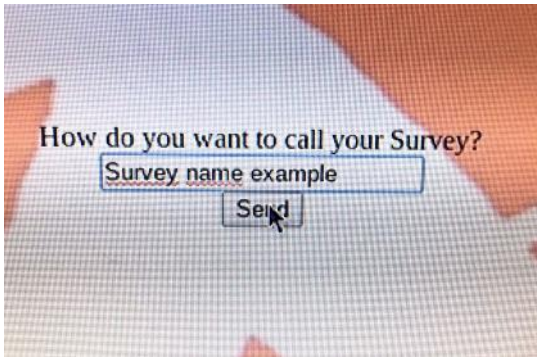


3. Once entered, we can select different option:

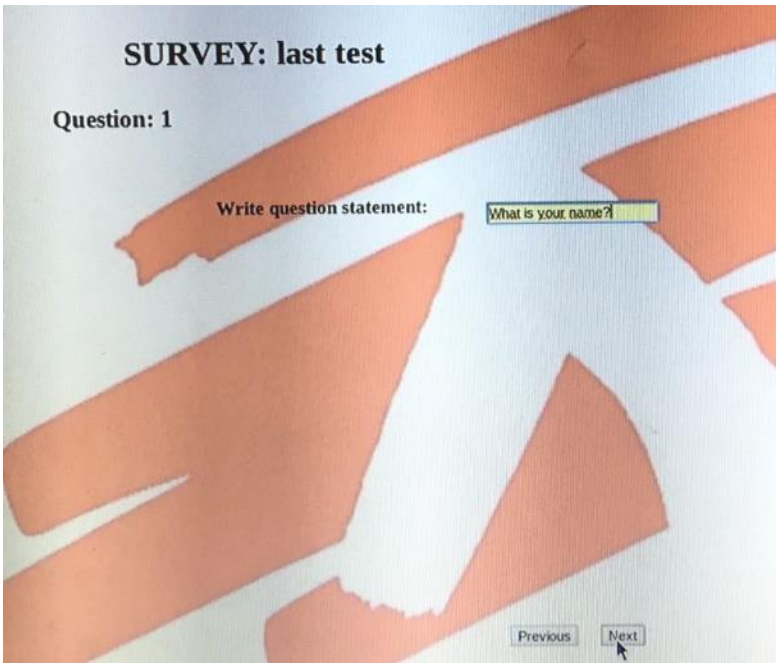
- Create a Survey
- Select Survey
- Answer Survey
- Edit Survey

We press Create survey

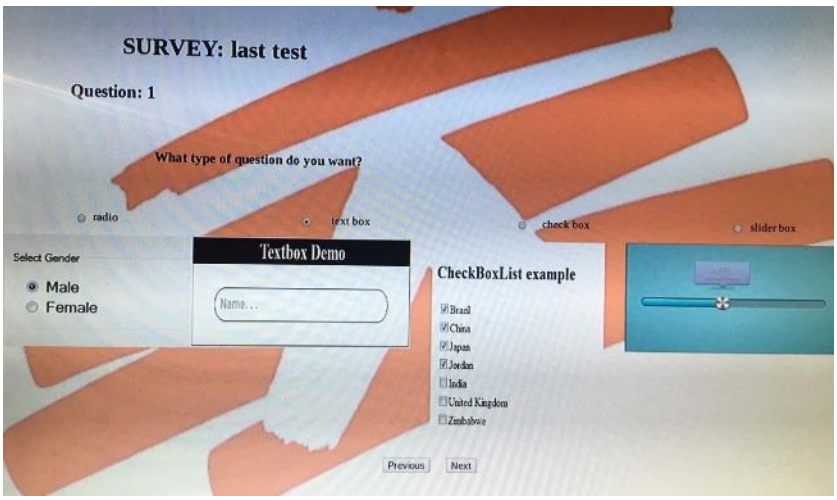
4. Enter the name of the survey you want to create. And Press Send



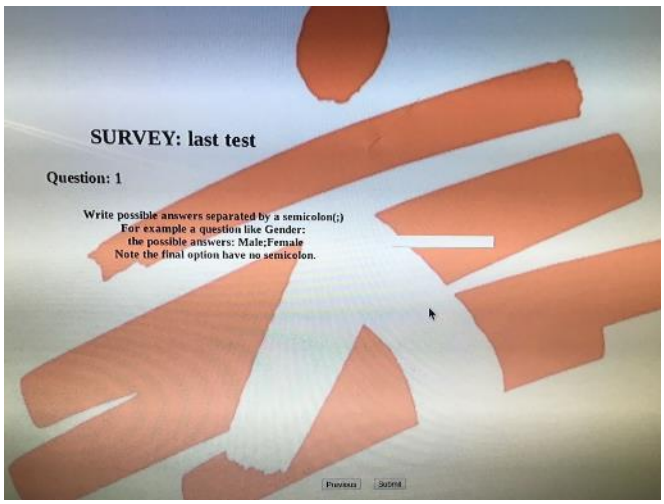
5. Write the question statement.



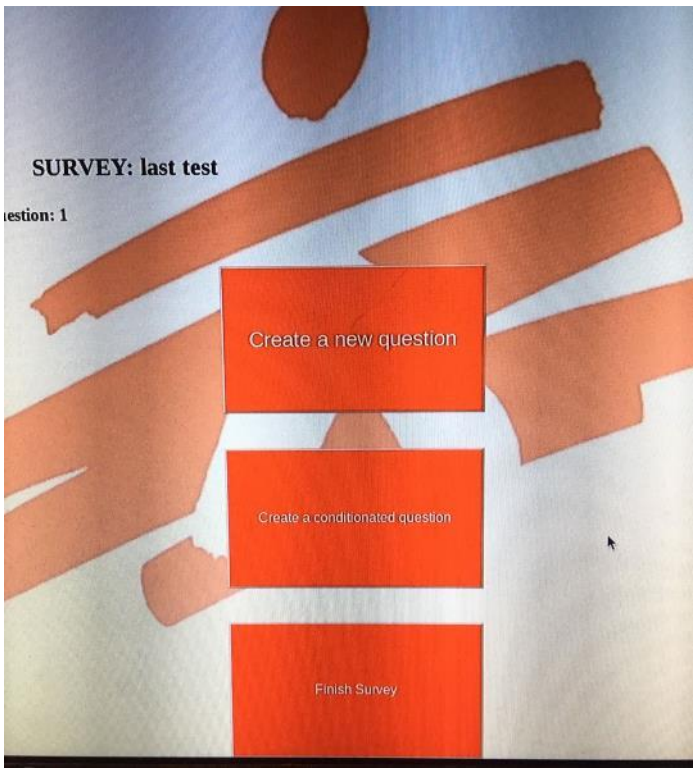
6. Select the question type. Then, press next



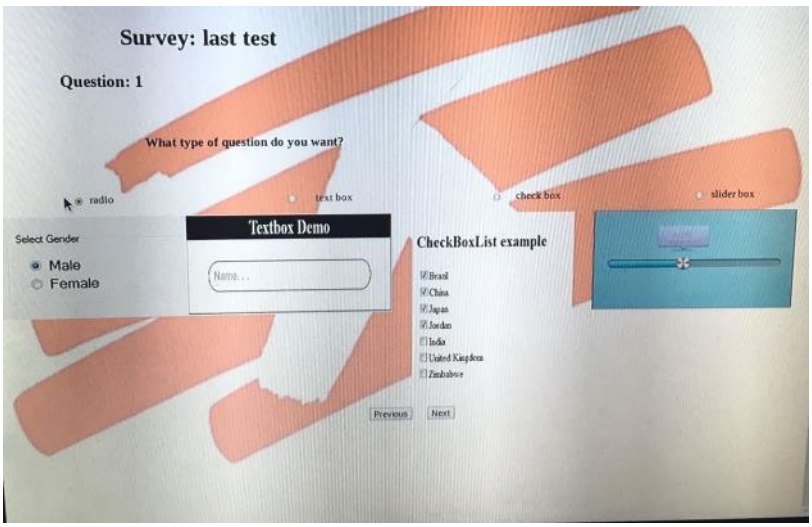
7. Enter the options of the closed-ended. question. Omit this stage if it is open-ended question. Finally Press Submit

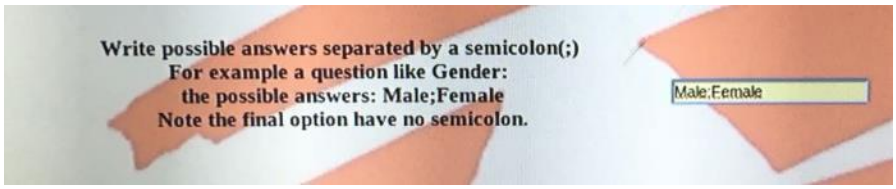


8. In the next, screen we have three options create a new question a conditionate question and finish survey. To create a conditioned question previously we must had entered a closed-ended question.

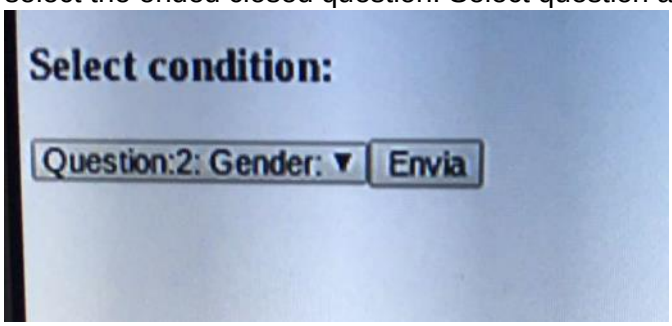


Example to create a closed-ended question:
Following the previous steps:

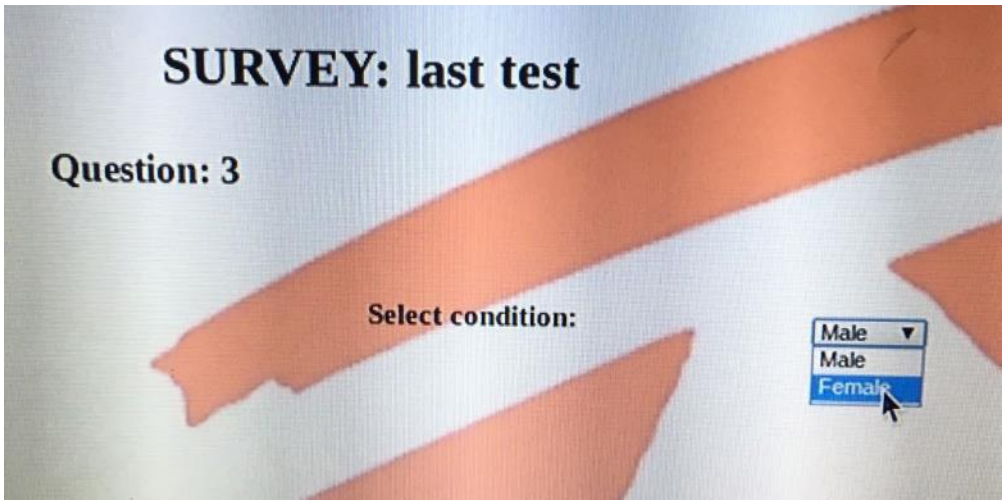




9. By clicking on create a conditioned question. It will ask for which question you want to create the conditioned one. So, there is a dropdown where it can only select the ended closed question. Select question and press Send.



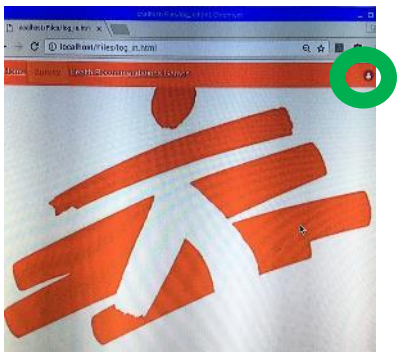
10. Select the condition to ask the question from the options of the previous selected question. Press Next



11. Like the other times, enter the question statement, type question and option if it is closed-ended question.

TUTORIAL TO EDIT A SURVEY

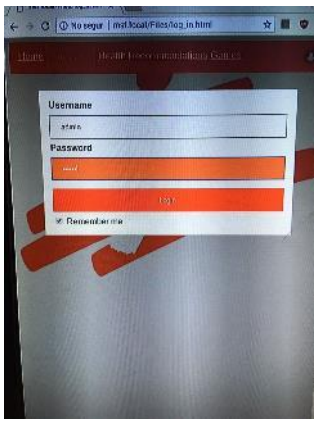
1. Once accessed to web, you must log in with administrator user. Press the icon on the right top corner of the page.



2. Register to the pop up form with the following credentials:

Username: admin

Password: admin

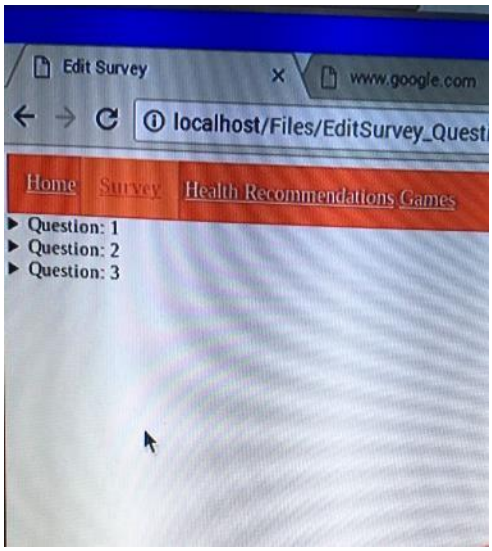


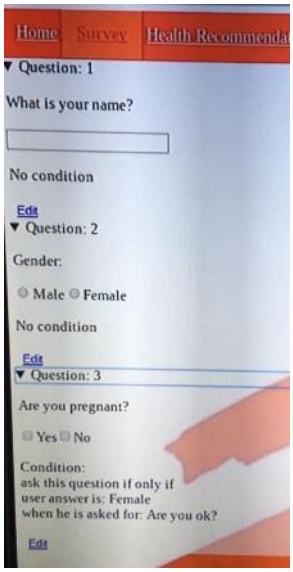
3. Once entered, we can select different option:

- Create a Survey
- Select Survey
- Answer Survey
- Edit Survey

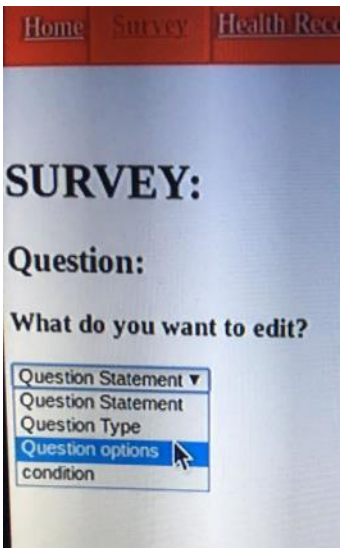
We press Edit survey

4. In the next screen, it will appear all the questions from the selected survey. By clicking, on the question it will display all the information. Click on Edit to edit the selected question.

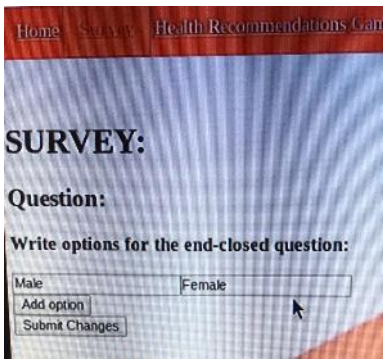


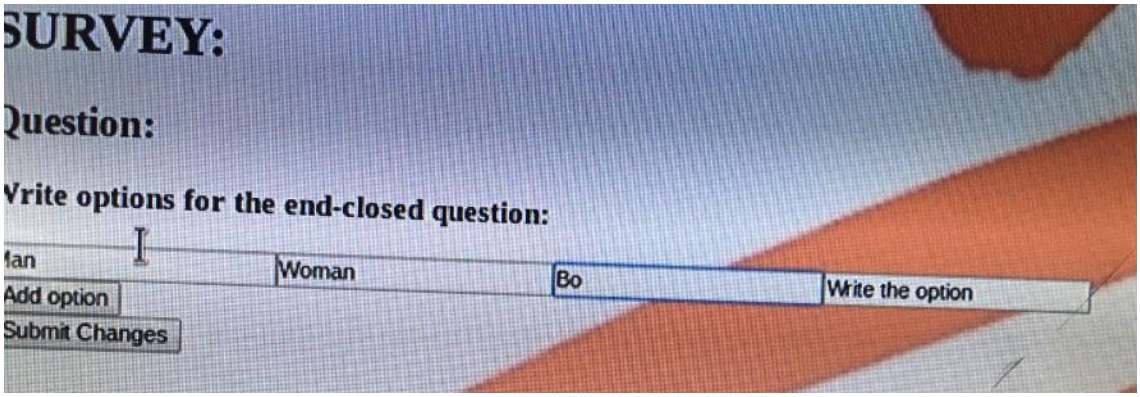


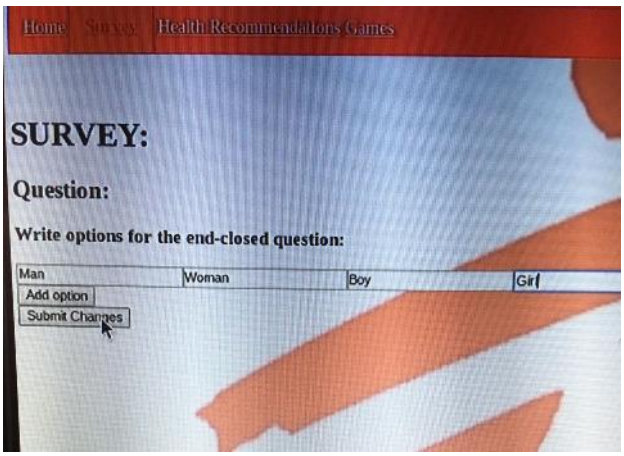
5. Select the part of the question that is desired to edit. And press Edit.



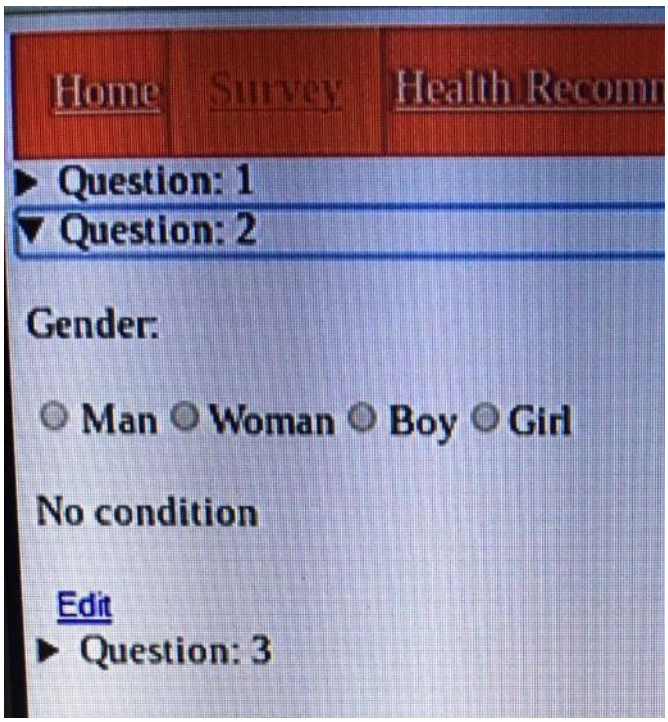
6. In the next screen are show the different question option. Pressin Add option, a new option and text box will be display. After editing all the text box as it is desired press Submit Changes.





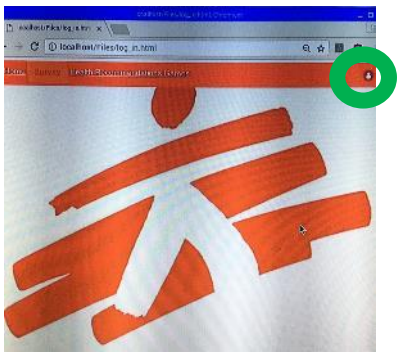


7. It will return to the previous page where chages done can be checked.



TUTORIAL TO SELECT A SURVEY

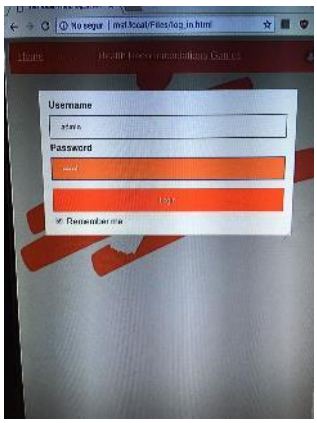
1. Once accessed to web, you must log in with administrator user. Press the icon on the right top corner of the page.



2. Register to the pop up form with the following credentials:

Username: admin

Password: admin

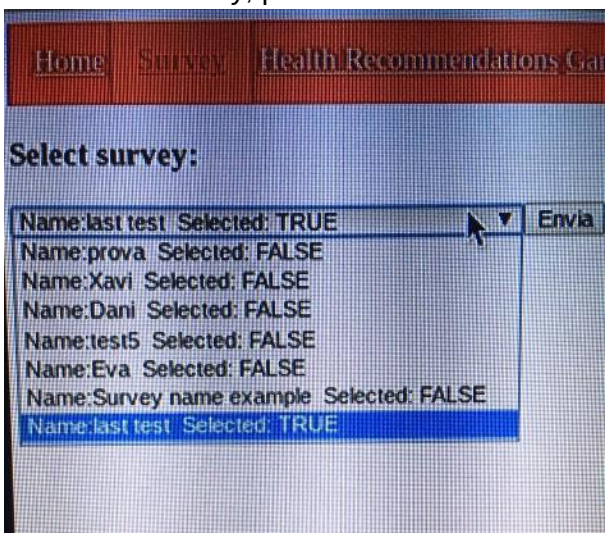


3. Once entered, we can select different option:

- Create a Survey
- Select Survey
- Answer Survey
- Edit Survey

We press Select survey

4. In the next screen, there is a dropdown where the survey can be selected. Once selected the survey, press Send.



1.1 Web Structure

The web's aim is to create a tool to generate Survey and answering them at the same time it provides multimedia resources to distract users.

We have structured the web site in two main different parts:

1. Administrative site (including new media, create survey)
2. Customer site.(visualization, answer survey)

1.1.1 Administrative site

Administrative site will be accessed by identifying user with administrative account. It will allow:

- Create survey.
- Edit Survey
- Select Survey
- Present Surveys results
- Introduce new media.

1.1.2 Customer Site

Custommet site will allow to:

- Answer Survey
- View media (videos, photos)
- Play some games

So, web will have 4 different parts:

1. Home
2. Health Recommendations
3. Survey
4. Games

First we will create a defult style for all web pages:

1.2 Code

1.2.1 Template

All documents in this project will have the following stucture

```
<!DOCTYPE html> <! --Instruction to browser about version HTML page is HTML5->
<html>  <! - - It starts html document - - >
        <head>
                <title> Create Survey <title>  <! - - browser tab name.
                                        It is useful for search-engine results - - >
                <meta charset="UTF-8"> <! - - Specify the character encoding for
                HTML document - ->
                <meta name="viewport" content="width=device-width,initial-
scale=1"> <! - -
                                        set width of the page to follow screen-width of
                                        the device with  the initial zoom level- - >
                <link rel="stylesheet" type="text/css"
href="CSS/background_nav.css">
                </link> <! - - To not copy the same style all the time we import a CSS
        file - ->
                <style></style>
        </head>
        <body>
                <header>
                <nav class="topnav"> <! - - It starts a navigation element - - >
                <a href="index.html" class="active">Home</a> <!- - defines hyperlink
                                        href specify the other server file and between
                                        opening an close tag (<a> name </a>) specify the
                                        text to display. Finally the class is to specify
                                        style of the element- - >
                <a href="Survey.html">Survey</a>
                <a href="HealthRecomendation.html">Health recommendation</a>
                <a href="games.html">Games</a>
                </nav>
                </header>
                <main>
                </main>
        </body>
</html>
```

The background_nav.css contains this:

```
/*The navigation object class*/
.topnav{
	overflow:hidden;
	background-color:red;
	padding:14px 16px;
	font-size:17px;
}
/*a is for style the hyperlynks (<a href="file_path">Name2display</a>)*/
.topnav a {
	color:white;
	text-decoration: none;
}
/*To style the hyperlynk element of the current page*/
.topnav a.active{
	background-color:tomato;
	color:red;
	padding:14px 16px;
	font-weight: bold;
}
/*To style the hyperlink when mouse is over them*/
.topnav a:hover{
	background-color:white;
	color:red;
	padding:14px 16px;
	font-weight: bold;
}
/*To style background page*/
html{
background-image:url("../Media/msf.jpg");
background-repeat: no-repeat;
background-size:cover;
}
body{
	background:rgba(255,255,255,0.5); //Apply transparency
}
```

1.2.2 Games

The easiest document is games.html. If we skipped the things explained above and go directly to the purpose of that file, we find the next code:

```
<div class="container">
<!- -  This allow to create a hyperlink by clicking on image.
First, we must include image tag <img> and
specify its source (image path where it is stored).
Next, we use JavaScript location.href function to redirect in onclick property of the image
Finally, we specify a class to style that image setting size position opacity… - -  >
<img src="Media/TicTacToe.jpg" class="centered" onclick="location.href='tic-tac-toe/index.html';">
<img src="Media/2048.jpg" class="centered_1" onclick="location.href='2048/index.html';">
</div>
```

In the style tag we include this CSS:

```
.centered{
position:absolute;
        top:25%;
        left:50%;
        height:25%;
        width:25%;
        transform: translate(-50%,-50%);
        display:block
}
.centered_1{
        position:absolute;
        top:60%;
        left:50%;
        height:25%;
        width:25%;
        transform: translate(-50%,-50%);
        display: block;
}
```

When users clicks to the image it will be redirect to the game's index.html file where he can play.

To add a game you must:

1. Download a free open code source
2. Extract the compressed files
3. Open the index.html file in your browser and try to play to check if everything works.
4. Move the folder to var/www/html/Files
5. Add and image with a link to the game's index.html in games html.

1.2.3 Videos

In order to make a dynamic web, and allow to an admin user with no programming skills insert videos. We must create a connection to database to present videos.Because, if we create a simple function that add a video when it reloads the page the video will not be presented. So, we must:

- Create function to add video in visualization page,
- Register that video to database.
- Store the video in the correct directory.

The following code is to present videos stored in the database in the correct directory.

To do that, it get the videos' path and categories to classify it and present into a details object and add it into a table.

```
<?php //open php

$servername = "localhost"; //declare and initialize a variable

$username = "admin";

$password = "admin";

$dbname = "dbSurvey";


// Create connection using MySQLi object

$conn = mysqli_connect($servername, $username, $password,$dbname);

// Check connection

if (!$conn) {

   die("Connection failed: " . mysqli_connect_error());

}

echo "Connected successfully"."<br><br>";
```

```
/*Populate data from database
To pass variables from PHP (database) to HTML we can follow the next steps:
1. Get the variable and stored to PHP
2. Echo into a div the variable
3. In HTML get the variable with document.getElementById*/

//CATEGORIES
//create a query to know how many categories we have stored
$sqlNumCat="SELECT COUNT(DISTINCT category)  FROM tbVideos";
//execute sql query with the function mysqli_query from MySQLi object
$result=mysqli_query($conn,$sqlNumCat);
//check for errors
if (mysqli_query($conn, $sqlNumCat) == TRUE) {
//get the result form the query. Stored in a array
$row = mysqli_fetch_assoc($result);
//declare and initialize a variable
$numberCat=$row['COUNT(DISTINCT category)'];
//passed to HTML by creating a div.
echo "<div id='numberCat'>".$numberCat."</div>";
} else {
echo "Error: " . $sqlNumCat . "<br>" . $conn->error;
}
//create a query to know the name of the possible categories.
$sqlCat="SELECT DISTINCT category  FROM tbVideos";
$result=mysqli_query($conn,$sqlCat);
if (mysqli_num_rows($result) > 0) {
        $i=0;
//look throught the array and echo the result into a div
        while($row = mysqli_fetch_assoc($result)) {
        echo "<div id=cat".$i.">".$row[category]."</div>";
        $i=$i+1;
        }
}
```

```
//ELEMENTS
//we repeat the same procedure done in categories to get elements.
$sqlNumberVideo="SELECT COUNT(id) FROM tbVideos";
$result=mysqli_query($conn,$sqlNumberVideo);
$row = mysqli_fetch_assoc($result);
$numberVideo=$row["COUNT(id)"];
echo "<div id='numberVideo'>".$numberVideo."</div>";

//aqui es fa una query que em donarà array categoria path
$sqlVideo="SELECT category,path FROM tbVideos";
$result=mysqli_query($conn,$sqlVideo);
if (mysqli_num_rows($result) > 0) {
        $i=0;
        while($row = mysqli_fetch_assoc($result)) {
        echo "<div id=".$i."_cat>".$row[category]."</div>";
        echo "<div id=".$i."_path>".$row[path]."</div>";
        $i=$i+1;
        }
}
?>

<! - -  Next, it follows the template explained before.>


< !- -In the script we create the details object and table object and we append the video - ->
```

```
<script>
// First we create the structure. We create the Details object and append Table to it details.

// Declare and initialize the category's number variable  from the database pass it by PHP
var catNumber=document.getElementById("numberCat").innerHTML;
var i;
// Look through categories
for (i=0;i<catNumber;i++){
var category = document.getElementById("cat"+i).innerHTML;
//Call  the function create details
createDetails(category);
//Call  the function append table
appendTable(category);
}
// Now the structure created we must append video
//Declare and initialize the video's number variable from the database pass it by PHP
videoNumber=document.getElementById("numberVideo").innerHTML;
var j
//Look through elements
for (j=0;j<videoNumber;j++){
videoCat=document.getElementById(j+"_cat").innerHTML;
videoPath=document.getElementById(j+"_path").innerHTML;
//Call function append video
appendVideo(videoCat,videoPath);
}

//This function creates a Detail object.
//A details object is an interactive box  which can be opened and closed by user. Details can contain any sort of content. Details includes a heading specified by <summary> tag
function createDetails(detailsName){
//Create element
var x= document.createElement("DETAILS");
```

```
//Specify id
x.setAttribute("id","d"+detailsName);
//Create Summary
var y= document.createElement("summary");
//Specify Summary
y.innerHTML=detailsName;
//Append Summary to details object
x.appendChild(y);
//Append details object to the HTML document
document.body.appendChild(x);
}

//This function creates and append table to specified details.
function appendTable(tableName){
//Create Table
var x = document.createElement("TABLE");
//Set Table id
x.setAttribute("id", "t"+tableName);
//Get Details
y=document.getElementById("d"+tableName);
//Append table to details
y.appendChild(x)
}

//This function appends video to a specified table
function appendVideo(tableName,path) {
//Get Table
  var table=document.getElementById("t"+tableName);
//Insert Row to the table
  row = table.insertRow(0);
//Insert Cell to the table
  cell1 = row.insertCell(0);
//Create video element
var video=document.createElement("VIDEO");
```

```
//Set video properties
video.src="Media/5pasitos.mp4";
video.width=300;
video.heigth=200;
video.controls=true;
//Append video to the cell
cell1.appendChild(video);
}
</script>
</body>
</html>
```

1.2.4 Survey

1.2.4.1 Create Survey

Create a survey implies two different tables tbSurvey (which stores all survey created) tbQuestions (which stores the suvey's questions).

For the first table we use the following code, we will need PHP to connect to database and insert the new Survey created and HTML to present and get user's input data:

```
<?php
//We skipped the part to connection to database because it's the same as before//
//INSERT DATA
/*When we submit HTML form, the name-value pairs are stored in the PHP global
variable $_POST. */
//check if the variable is set
if ($_POST[nameSurvey]!=NULL){
/*Create SQL statement to insert into tbSurvey the new Survey.
NOTE: Thanks to tbSurvey definition it is not necessary to introduce other fields.
Id is auto incremental field and selected the default value for new records is true.*/
$sql= "INSERT INTO tbSurvey (nameSurvey) VALUES
('" .$_POST[nameSurvey]."')";
//Create a query to set all the different Survey selected to false
$sql_set_selected="UPDATE tbSurvey SET selected=FALSE WHERE
                        nameSurvey!='".$_POST[nameSurvey]."'";
        if (mysqli_query($conn, $sql) === TRUE) {
                        if (mysqli_query($conn, $sql_set_selected) === TRUE) {
//                      echo "Selected survey changed";
                        /*Once the Survey is created we redirect to the next form to fill
the
                        survey's questions*/
                        header('Location:CreateSurvey_3.php');
                        }else{
                        echo "Error: " . $sql_set_selected . "<br>" . $conn->error;
                        }
        } else {
                echo "Error: " . $sql . "<br>" . $conn->error;
        }
}
?>
```

<! - - If we skip the parts explained on the template. -->

<! - - A form with HTML is introduced with the tag <form>.

Action is a required attribute and specify what does the form when it is submitted. Action specify form handler (a server page with a script which process input data).

In this case, it is the PHP of the same page who connect to database and store user's input.

Method attribute specify the HTTP method (GET or POST) to be used. So, PHP will get the data in $_GET or $_POST variable.- - >

```
<form class="centered" action="<?php $_PHP_SELF ?>"  id="myForm" method="POST">
How do you want to call your Survey?<br>
<input type="text" name="nameSurvey"><br>
<input type="submit" value="Send">
</form>
```

To finish the creation of survey we must generate question. This is done with the next code:

The principal part of this code is the next form:

```
<! - - We use a class tab to  create step by step form and not generate a form with all the question in one page.
Styling tab with CSS and JS it will show only one question. - - >
<form id="regForm" action="CreateSurvey_2_form.php" method="POST">
<! - -First Tab. Buttons: 1)Create new question. 2)Conditioned question. 3)Finish Survey  - - >
        <div class="tab" id="question">
                <button type="button"
                        id="NotCondBtn"
                        class="centered_btn"
                        onclick="condQuestion(0)">
                                Create a new question
                </button>
                <button type="button"
                        id="condBtn"
                        class="centered_btn_1"
                        onclick="condQuestion(1)" >
                                Create a conditionated question
                </button>
                <button type="button"
```

```
                id="finishSurvey"
                class="centered_btn_2"
                onclick="condQuestion(2)" >
                        Finish Survey
            </button>
        </div>
<! - - Second tab. Select (Dropdown/Combo box) of the options you want to create
conditioned question- - >
        <div class="tab" id="last_option">
            <h3 header class="centered_header_3"
                id="h_option">
                        Select condition:
             </h3>
            <select name="mycondition"
                class="text"
                form="regForm"
                id="condSelect">
            </select>
        </div>
<! - - Third Tab. Text to write question's statement. - - >
        <div class="tab" id="statement">
            <h3 header class="centered_header_3"
                id="h_statement">
                        Write question statement:
            </h3>
            <input type="text" name="statement" class="text">
        </div>

<! - - Forth tab. Radio button with example image to select question's type - ->
        <div class="tab" id="type">
            <h3 header class="centered_header_3" id="h_type"> <! - - Title - ->
                                What type of question do you want?
            </h3>
            <div>  <!--First radio option. Specify name to the  name-value pairs of
post -->
```

```
                <input type="radio" name="type" id="radio" class="radio1"
                                                                value="radio">
                        <label for="radio"> <! - - Label for first radio button - - >
                                <p class="radio1_label"> radio </p> <!- - Text - ->
                                <! - - Image - - >
                                <img
                        class="radio1_img"src="Media/radiobutton.jpg”>
                                                                        </img>
                        </label>
                </input>
        </div>
        <div> <! - - Second radio option- ->
                <input type="radio" name="type" id="text" class="radio2"
                                                                value="text">
                        <label for="text">
                                <p class="radio2_label"> text box</p>
                                <img class="radio2_img"
src="Media/textbox.jpg" >
                                                                        </img>
                        </label>
                </input>
        </div>
        <div><! - - Third radio option- ->
                <input type="radio" name="type" id="check" class="radio3"
                                                        value="check">
                        <label for="check">
                                <p class="radio3_label"> check box</p>
                                <img class="radio3_img"
src="Media/Checkbox.jpg" >
                                                                        </img>
                        </label>
                </input>
        </div>
        <div><! - - Forth radio option- ->
                <input type="radio" name="type" id="slider"class="radio4"
```

```
                                                                    value="slider">
                        <label for="slider">
                                <p class="radio4_label"> slider box</p>
                                <img class="radio4_img" src="Media/slider.jpg"
>
                                                                                    </img>
                        </label>
                    </input>
                </div>
            </div>

<! - - Fifth tab. Text to write the options of a closed-ended question - - >
            <div class="tab" id="option">
                <h3 header class="centered_header_3" id="h_option">
                    Write possible answers separated by a semicolon(;)<br>
                    For example a question like Gender: <br>
                    the possible answers: Male;Female<br>
                    Note the final option have no semicolon.
                </h3>
                <input type="text" name="option" class="text2">
            </div>
<! - - Buttons (Next/Previous) to navigate through tabs - - >
            <div class="buttonForm">
                    <button type='button' id='prevBtn' onclick='nextPrev(-1)' >
                    Previous
                    </button>

                    <button type='button' id='nextBtn' onclick='nextPrev(1)'>
                    Next
                    </button>
            </div>
<! - - Necessary database input but introduced transparent for user - ->
            <div>
            <input type="hidden" name="idSurvey" value=999 id="idSurvey" > <br>
```

```
        </div>
        <div>
        <input type="hidden" name="idQuestion" value=999 id="idQuestion"> <br>

        </div>
        <div>
        <input type="hidden" name="idCondition" value=999 id="idCondition">
<br>
        </div>
</form>
```

To make possible this interaction explained we need this script

```
<script>
//We use the same methodology to pass data from PHP to JS. It means with PHP we echo the variables into a div and get the div with document.getElementById


//We search for the previous idQuestion stored in database from the same survey (from the same idSurv). If there is no one (idQuest=Null) id Question is set to 0.


/*if it is the first question of the Survey we must skip the
- First tab (action tab):
    o Create a new question
    o Create a conditioned question
    o Finish the Survey
- Second tab (Condition tab): select the condition from the possible options of closed-ended question that do you want to create a conditioned question.
  For Example:
  - Gender:
      o Male
      o Female
  - Are you pregnant ? (ask conditioned question only when the previous answer is equal to Female).

  So possible condition to select in a conditioned question of first question are :

  Male Female.

  Admin user will have to select Female to generate the question 2 properly.*/


if(QuestId==0) {
        QuestId=1; //Set the idQuestion to store in database
        CondId=0; //Set the idCondition to store in database
        currentTab=2;//Set current tab
        showTab(2,1); // Go to the third tab (0,1,2,…)call function to show the current tab
}
//If the next question is a conditioned question we must go to the second tab, skipping the first one.
else if(CondQuestId!=0){
        QuestId=QuestId+1;
        currentTab=1;
        showTab(1,1);
}
```

```
//We must go to the first tab the rest of possibilities.
else{
currentTab=0;
showTab(0,1);
}
<! - - For better visualization experience, we add Survey name and actual question -
- >
document.getElementById('header').innerHTML="SURVEY: "+SurvName;
document.getElementById('question').innerHTML="Question: "+QuestId;



//We fill the selector with the option obtained from the database passed through
PHP
sel=document.getElementById("condSelect");
array=optionList.split(";")//split the string with a semicolon as a delimiter

for (i=0; i<=array.length-1;i++){
        option =document.createElement('option'); //create option element
        option.value=array[i]; //Set option value
        option.text=array[i]; //Set option text
        sel.add(option); //add option element into the selector
}
/*This function is to pass the Survey, Question and Condition's id value set in JS in
to HTML, to pass them into PHP to store in database. */
 function submitForm(){
        //Get element to set value
        document.getElementById("idSurvey").value=SurvId;
        document.getElementById("idQuestion").value=QuestId;
        document.getElementById("idCondition").value=CondQuestId;
        //Submit form
        document.getElementById("regForm").submit();
}
/*This function is used for fist tab which allow this functionalities:
    ○   Create a new question
    ○   Create a conditioned question
    ○   Finish the Survey
```

```
*/
function condQuestion(newQuest){
if(newQuest==1){//1 is to create a new conditioned question
//we redirect to other file to get condition from options of the question that user will choose
        window.location="SelectQuestion.php";

}if(newQuest==0){//0 is to create a new question
        CondId=0; //we set condition id to 0.
        var x = document.getElementsByClassName("tab");
        x[1].bloqued=1;//we block the second tab (where you select condition)
        QuestId=QuestId+1;
}if(newQuest==2){//2 is to finish the survey
        window.location="index.html";//redirect to home page.
}
document.getElementById('header').innerHTML="Survey: "+SurvName;
document.getElementById('question').innerHTML="Question: "+QuestId;
nextPrev(1);//Go to the next tab
}
//this function is to change previous and next button and skip a tab is blocked
function showTab(n,direction) {
  var x = document.getElementsByClassName("tab");
        while(x[n].bloqued==1){//find the first tab not bloqued
                if(direction==1){//if direction is forward we must add 1 to go to the next tab
                n=n+1;
                currentTab=n;
                }
                else{//if direction is backward we must subtract 1 to go to the next tab
                n=n-1;
                currentTab=n;
                }
        }
  x[n].style.display = "block";//make tab visible
        if (n == 0) {//if it is the first tab we hide next and previous button
```

```
        document.getElementById("prevBtn").style.display = "none";
        document.getElementById("nextBtn").style.display = "none";
        } else {//otherwise we show next and previous button
        document.getElementById("prevBtn").style.display = "inline";
        document.getElementById("nextBtn").style.display = "inline";
        }
        if (n == (x.length - 1)) {//if it is the last tab we change the next button to
Submit
        document.getElementById("nextBtn").innerHTML = "Submit";
        } else {//otherwise the next button will display next
        document.getElementById("nextBtn").innerHTML = "Next";
        }
}
// This function will figure out which tab to display
function nextPrev(n) {
   var x = document.getElementsByClassName("tab");
 // Hide the current tab:
  x[currentTab].style.display = "none";
  // Increase or decrease the current tab by 1: n could be positive or negative
  currentTab = currentTab + n;
  // if you have reached the end of the form...
  if (currentTab >= x.length) {
   //  the form gets submitted:
  submitForm();
  }
  // Otherwise, display the correct tab:
  if(n==1){//forward direction
  showTab(currentTab,1);
  }else{//backward direction
  showTab(currentTab,0);
  }
}
</script>
```

The previous form uses a PHP file to submit the form.

```
<?php
$servername = "localhost";
$username = "admin";
$password = "admin";
$dbname = "dbSurvey";
// Create connection
$conn = mysqli_connect($servername, $username, $password,$dbname);
// Check connection
if (!$conn) {
   die("Connection failed: " . mysqli_connect_error());
}
//echo "Connected successfully"."<br><br>";

//INSERT DATA
//create a query to insert input data to database.
$sql= "INSERT INTO tbQuestion
(idSurvey,idQuestion,statement,option,mycondition,idCondition,type)
VALUES (" .$_POST[idSurvey]. " , ".$_POST['idQuestion'].
" ,'" .$_POST["statement"]. "' , '" .$_POST["option"]."' , '" .$_POST["mycondition"].
"' , " .$_POST['idCondition']." , '" .$_POST["type"]. "' )";

if (mysqli_query($conn, $sql) === TRUE) {
   echo "New records created successfully";
//Once the form it is submitted we must redirect to add the next question
header("Location: CreateSurvey_3.php")
} else {
echo "Error: " . $sql . "<br>" . $conn->error;
}
?>
```

When we want to create a conditioned question, the question relating to you want to create this new conditioned question, must be asked to user. To do this, in the JavaScript we redirect to another form to grab the question that user wants to ask the question related.

The main importance code of this form is the following:

We have opened php connected to database and get stored the current survey (Selected survey) in $SurvId variable.

```
//We will do the next query to get all end-closed questions from the current survey means it contains options. If they do not content option we cannot create a condition based on user previous answer.

$sqlquestion="SELECT idQuestion,statement FROM tbQuestion WHERE idSurvey=".$SurvId. " AND option!='';
```

We have opened html tag and follow the template and on body tag we open again php.

```
//if previous query have results

if (mysqli_num_rows($result) > 0) {

echo "<form id='conditionForm' action='$PHP_SELF' method='POST'>";//Using echo we write html and create form

echo "<div id='option'>"; //Using echo we write html and create a div

echo "<h3> Select condition:  </h3>";//Using echo we write html and create a Header

echo "<select name='question' id='condSelect'form='conditionForm'>";  //Using echo we write html and create a select

//while the query get results we go through query rows

while($row = mysqli_fetch_assoc($result)) {

        $QuestId=$row['idQuestion'];//Store question id number

        $statement=$row['statement'];//Store question statement

echo "<option value=".$QuestId.">Question:".$QuestId.": ".$statement."</option></div>";

//Using echo we write html and create option (dropdown) with the statement and id.

}

}

echo "<div><input type='submit'></div>";//Using echo we write html and create submit button

echo "</form>";//we close form

$_SESSION['idQuestion']=$_POST['question']; //We will use global SESSION array to pass the selected question to other files
```

```
if(isset($_SESSION['idQuestion'])){//Once the variable is set
header("Location: CreateSurvey_3.php");//We redirect to the main from to create
                                                                                survey
}
?>//Close PHP
</main><!--Close main-->
</body><!--Close body-->
</html><!--Close html-->
```

1.2.4.2 Edit Survey

To edit a created survey, we will use the following code:

We have followed the templated and in body tag we have opened the php connected to the database and stored in variables:

- Current survey id from
- Current survey name
- Current survey question number.

Finally, we do a query to get all the current survey questions

```
if (mysqli_num_rows($result)>0){ //If the query has result
        while($row=mysqli_fetch_assoc($result)){//while the query has result we will go through query rows
//By using echo we write html and…
            echo "<details> //open details
                <summary>Question: ".$row["idQuestion"]."</summary> //Create Summary element
                    <table id=".$row["idQuestion"]."> //Create Table
                      <tr> //Create table row
                          <td>//Create row cell
                              <p>".$row["statement"]." </p>//Create a paragraph and add statement question
                          <td>//Close row cell
                      </tr>//Close table row
                      <tr>//Open table row
                         <td>";//Open row cell. Close php echo function because we
        need to use php's function
switch ($row['type']){//Generate the response depending on the type
                case "radio": //In case is a radio button type
                                $str =explode(';',$row["option"]);//Get the option array
        $i=0; //To go through the array we set the first position by declaring and
initializing variable to 0
                $length=sizeof($str)-1;//Set the last position
                for($i; $i <= $length; $i++){//loop through the array
                                echo "<input type='radio' //Create a radio button
                element
                                name='answer".$row["idSurvey"] //naming the radio
                            button with survey id
                                .$row["idQuestion"]//and question id
```

```
                        .$row["idCondition"]//and condition id
                        ."' value=$str[$i]>//setting radio button default value
                        $str[$i]";//set a label to radio button
                }
                break;//end case
        case "check"://repeat the same as radio button case with checkbox
                $str =explode(';',$row["option"]);
                $i=0;
                $length=sizeof($str)-1;
                        for($i; $i <= $length; $i++){
                                echo "<input type='checkbox'
                                        name='answer".$row["idSurvey"]
                                        .$row["idQuestion"]
                                        .$row["idCondition"]
                                        ."' value=$str[$i]">"
                                        .$str[$i];
                        }
                break;
        case "slider"://repeat the same as radio button case with slider
                echo "<input id='move_range' type='range'
                        name='answer".$row["idSurvey"]
                        .$row["idQuestion"]
                        .$row["idCondition"]
                        ."' min='0' max='100' step='5' value='50'>";
                echo "<span id='move_value'>50</span>"; //incorporate slider value
text
                break;
        case "text":
                echo "<input type='text'
                        name='answer".$row["idSurvey"]
                        .$row["idQuestion"]
                        .$row["idCondition"]."'><br>";
        break;
        default:
        break;
}
```

```
                echo "</td>//close row cell
                    </tr>//close table row
                     <tr> //open table row
                         <td>";//open table. Close php echo function because we need
                to use  php's function
if($row[idCondition]!=0){ //If question have a condition
   $sql2="SELECT statement FROM tbQuestion WHERE
       idQuestion=".$row[idCondition];  //Query to select the conditioned question
statement
   $result2 = mysqli_query($conn,$sql2); //Execute Query

   $row2  = mysqli_fetch_assoc($result2);//get results
                            echo "<p>Condition:<br> //write paragraph

                                ask this question if only if<br>
                                user answer is: ".$row[mycondition]."<br> //add
                        condition
                                when he is asked
                        for:".$row2[statement]."</p>";//statement
}else{  //if have no condition

                        echo "<p>No condition</p>";//write paragph

}

                    echo"</td> //Close row cell
                        </tr>//Close table row
                         <tr>//Open table row
                          <td>//Open row cell
                                <a href='CreateSurvey_3_edit.php'>Edit</a>//Insert
lynk
                            </td>//Close row cell
                        </tr>//Close table row
                </table>//Close table
            </details>";//Close details
            }
}
```

Answer Survey

To create a dynamic survey we must populate question from admin user input stored into the database.

Skipping the less important things we get to the following code:

```
//In $SurvId we have stored the survey that it is selected.

//Create a query to get questions of selected survey

$sql = "SELECT * FROM tbQuestion where idSurvey=".$SurvId." order by case when idCondition=0 then idQuestion when idCondition!=0 then idCondition END ASC";

$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {

        while($row = mysqli_fetch_assoc($result)) {

                //we create a tab (div) with the id Condition id and condition string

                echo  "<div class='tab' id=$row[idCondition]_$row[mycondition]>"

                        .$row["statement"]; //we print the question statement

                switch ($row["type"]){//Generate a question depending of the stored type

                        case "radio":

                                $str =explode(';',$row["option"]);//Get the option array

                                $i=0;

                                $length=sizeof($str)-1;

                                        for($i; $i <= $length; $i++){//Look through the array

                                                echo "<input type='radio' //Set input type

/*when a form that use method POST is submitted a name-value pair are stored in $_POST array. PHP have associative array (arrays where the index are string).*/

                                                name='answer" //Set index POST array name

                                                .$row["idSurvey"] //adding Survey into index

                                                .$row["idQuestion"] //adding question into
                                                                                //index

                                                .$row["idCondition"] //adding Condition into
                                                                                //index

                                                ."' value=$str[$i] //Set Value of radio button
```

```
                    onclick=\"jumpTab('".$str[$i]."');\">// Set
                    //onclick property to jump conditioned
            question
                    $str[$i]"; //Print the option
            }
        echo "</div>";        //close tab
        break;
//The same for the other case of the switch
        case "text":
            echo "<input type='text'name='answer"
                .$row["idSurvey"]
                .$row["idQuestion"]
                .$row["idCondition"]."'><br>";
        echo "</div>";
        break;
        case "check":
            $str =explode(';',$row["option"]);
            $i=0;
            $length=sizeof($str)-1;
                for($i; $i <= $length; $i++){
                echo "<input type='checkbox' name='answer"
                    .$row["idSurvey"]
                    .$row["idQuestion"
                    ].$row["idCondition"]
                    ."' value=$str[$i]
                    onclick=\"jumpTab('".$str[$i]."');\">"
                    .$str[$i];
                }
        echo "</div>";
        break;
        case "slider":
        echo "<input id='move_range' type='range' name='answer"
                .$row["idSurvey"]
                .$row["idQuestion"]
                .$row["idCondition"]
```

```
                                    ."' min='0' max='100' step='5' value='50'>";
                                echo "<span id='move_value'>50</span>";
                        echo "</div>";
                        break;
                        default:
                        echo "</div>";
                        break;
                }
        }

//Then there the same button to the other form to navigate and go to the next tab
//Once the form is submitted, we must store data to the database.

//method POST save on $key the string name of the form and on $answer the user input
        foreach ($_POST as $key => $value){
//we have stored into $_POST index name answer+idSurvey+idQuestion+idCondition
//For example: answer110
        //str_split splits string into array with length of each array.
        //htmlspecialchars converts '<' and '>' to HTML entities
        $str= str_split(htmlspecialchars($key),6); //input answer110
                                                   //result [0]=answer [1]=110
        $answer=htmlspecialchars($value);//user input
        $surv=str_split($str[1],1);//input 110  result [0]=1;[1]=10
        $idSurv=$surv[0];//input [0]=1
        $quest=str_split($surv[1]);//input 01 result [0]=1;[1]=0
        $idQuest=$quest[0];//input [0]=1
        $idCond=$quest[1]; //input [1]=0
        //Create SQL statement to insert data to databse.
        $sql= "INSERT INTO tbAnswers (idSurvey,idQuestion,idCondition,answer)
        VALUES (" .$idSurv. " , " .$idQuest. " , " .$idCond. " ,'".$answer."')";
                if (mysqli_query($conn, $sql) === TRUE) {
                //    echo "New records created successfully";
                } else {
```

```
            //    echo "Error: " . $sql . "<br>" . $conn->error;
            }

        }
}
```

One more function in this html file is for slider. Slider is a HTML5 element and not all browser includes a number that changes when user move the slider. To emulate this we have the following function:

```
//get the span and we add event listener when it moves it will run the function.
document.getElementById("move_range").addEventListener('change',function(){
//get the span
l=document.getElementsByTagName("span");
for (i=0;i<=l.length;i++){
//set the value to the span
l[i].innerHTML=this.value;
}
},false);
```

Finally, we have the same functions to move through tabs

1.2.4.3 Select Survey

Now user is able to create a new survey, answer and edit current survey. Thus, web server must offer the possibility to change the selected survey. This code is very likely to select question used to ask which question user wants to create a conditioned question.

We have followed the templated and in body tag we have opened the php connected to the database. Finally, we do a query to get all the survey.

```
//Using echo we write html and…
echo "<form id='surveyForm' action='$PHP_SELF' method='POST'>";//create a form
echo "<div id='option'>";//create a div
echo "<h3> Select survey:  </h3>";//create a Header
echo "<select name='survey' id='survSelect'form='surveyForm'>";//create a select
                                                        (dropdown or combobox)
while($row = mysqli_fetch_assoc($result)) {//while the query has result we will go through query rows
        $idSurvey=$row['idSurvey'];//store survey's id
        $nameSurvey=$row['nameSurvey'];//store survey's name
        $selected=$row['selected'];//store  survey's selected attribute
        if ($selected==1){//if it is the current survey
                $selected="TRUE";//selected is true
        }else{//if it is not the current survey
        $selected="FALSE";//Select to false
        }
        echo "<option value=".$idSurvey." selected>//create a select's option element
                                Name:".$nameSurvey //Naming the select's option
                                ."    Selected: ".$selected."//Set as default value
                </option></div>";//Close select's option element
}//Closing while
}//Closing if
echo "<div><input type='submit'></div>";//Create submit button
echo "</form>";//Close Form
if(isset($_POST['survey'])){//When form is submitted check if POST['survey'] is set
$sql_set_false="UPDATE tbSurvey SET selected=FALSE";//Set all Survey not selected
$sql_set_selected="UPDATE tbSurvey SET selected=TRUE WHERE idSurvey=
```

```
                    ".$_POST['survey']."";//set users selection as selected/current survey
        if (mysqli_query($conn, $sql_set_false) === TRUE) {//Check if query execution works
                if (mysqli_query($conn, $sql_set_selected) === TRUE) {{//Check if query execution works
                        //echo "Selected survey changed successfully";
                        header('Location:Survey.html');//redirect to Survey home page
                }else{//if query execution do not work echo the error
                        echo "Error: " . $sql_set_selected . "<br>" . $conn->error;
                }//Close if
        } else {//if query execution do not work echo the error
                echo "Error: " . $sql . "<br>" . $conn->error;
        }//Close if
}//Close if
?><! - - Close php and in the following lines it closes body tag and html tag - - >
```

9 Glossary

MSF: *in French, stand for* Médecins Sans Frontières

UPC: *in Catalan, stand for:* Universitat Politècnica de Catalunya.

AUCOOP *in Catalan, stand for*: Associació d'Universitaris per la COOPeració

ETSETB *in Catalan, stand for:* Escola Tècnica Superior d'Enginyeria de Telecommunicacions de Bardcelona

ESADE *in Catalan, stand for:* Escola Superior Administració i Direcció d'Empreses

NGO *stands for:* Non Governmental Organization

SAME *stands for:* Severe Medical Adverse Event: defined as an injury resulting in prolonged hospitalization, disability or death, caused by healthcare management

WWW *stands for:* World Wide Web

HTTP *stands for:* Hyppertext Transfer Protocol

HTML *stands for:* Hypertext Mark-up Language

URL *stand for:* Uniform Resources Locators

W3C *stand for:* World Wide Web consortium

XHTML *stand for:* Extensible Hypertext Mark-up Language

CSS *stand for:* Cascading Style Sheet

JS *stand for:* JavaScript

DOM *stand for:* Document Object Model

PHP *stands for*: PHP Hypertext Pre-processor

PDO *stand for:* PHP Data Objects)

SQL *stand for:* Structured Query Language

MySQL *stand for:* "My" name of cofounder daughter and SQL

MySQLi *stand for:* MySQL Improve

RDBMS *stand for:* Relational Database Management System

SBC *stand for:* Single Board Computer

AP *stand for:* Access Point

NAT *stand for* Network Address Translation

LAMP *stand for* Linux Apache MySql PHP is archetypal model of web service stacks.