

Bachelor's Thesis (UAS)
Information Technology
2012

Stanislav Moyiseyev

CREATING POLLS ON MOBILE DEVICES



TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

BACHELOR'S THESIS (UAS) | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Degree programme: Information Technology

Winter 2012 |46

Supervisor: Patric Granholm

Stanislav Moyiseyev

Abstract

In the modern world the mobile phone has become an indispensable part of our everyday life. Nowadays it seems like the babies are born with mobile phones in their hands but not the silver spoon in their mouth like it used to be said.

The aim of this thesis is to create a poll application on mobile phones using PHP, JQuery, XML and JQuery Mobile. Using PHP and XML together, the author creates a database first and implements questions with associated answers for polls. Finally, and at the end there will be the chart of answers and see which answers have the highest percentage.

KEYWORDS:

polls, mobile devices, JQuery Mobile, charts, PHP, XML, HTML, JQuery, Ajax.

TABLE OF CONTENTS

1. INTRODUCTION	5
2. PROGRAM OVERVIEW	6
1. 2.1 AJAX	8
2. 2.2 DOM	9
3. 2.3 HTTP	10
4. 2.4 SQL	11
5. 2.5 XML	12
6. 2.6 JQuery	13
7. 2.7 PHP – (Hypertext Preprocessor)	14
3. GENERATING CODE FOR POLLS	16
3.1 Creating data code for MySQL	16
3.2 Creating PHP pages	17
3.3 Creating answers PHP page	19
3.4 Generating answers PHP page	19
3.5 The voting PHP page	19
3.6 The polls PHP page	20
3.7 Creating the front	20
4. SUMMARY	
Figures	
Figure 1. SQL Sample Script	6
Table 1. PHP Script	7
Figure 2: AJAX Sample	7
Figure 3: The hierarchy of DOM (Document Object Model)	9
Figure 4: HTTP example	10
Table 2: Basic XML example	12
Figure 5: JQuery example	13
Table 3: Sample PHP code	15
Figure 6: Adding questions	18

Figure 7: Home web page	21
Figure 8: Answers PHP page	22
Figure 9: Results page	23
Figure 10: Results with multiple votes	24
Figure 11: JQuery web page for mobile device	43

APPENDICES

Appendix 1.1 Record 1: database
Appendix 1.2 Code for storing answers into the database
Appendix 1.3 HTML syntax code
Appendix 1.4 Record 2: Question page
Appendix 1.5 Record 3: Creating answers PHP
Appendix 1.6 Record 4: Generating answers PHP
Appendix 1.7 Record 5: The voting PHP page
Appendix 1.8 Record 6: The polls PHP page
Appendix 1.9 Record 7: The front page html
Appendix 1.10 Testing the JQuery Mobile application HTML code

List of Abbreviations (OR) Symbols

AJAX	Asynchronous JavaScript
DOM	Documented Object Model
HTTP	Hypertext Transfer Protocol
SQL	Structured Query Language
XML	Extensible Markup Language
CSS	Cascading Style Sheets
PHP	Hypertext Preprocessor

1. INTRODUCTION

The power of mobile devices today is amazing. Nowadays mobile technologies attract plenty of interests in the modern world. They are becoming more powerful and smarter than computers a few years back. They come with dual core processors to handle heavier applications. Web browsing is one of the main uses of the mobile phones today. With tool like JQuery Mobile it is easy to create many different applications.

Using PHP and XML together in the HTML background with JQuery and JQuery Mobile in the foreground it is easy to create an interactive poll for a mobile device. By implementing to the web server an interactive poll application, an administrator can easily create new questions with answers so that users can navigate through the questions and chose the best or the answers of their choice. After answering the polls, at the end a graph of the results can be seen.

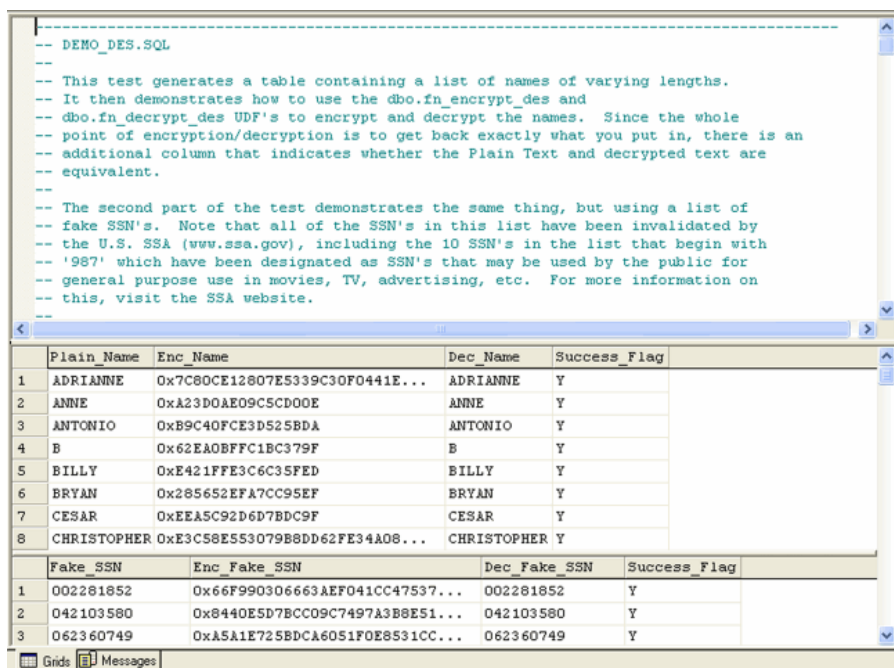
Starting with the SQL database, code for polls will be created using MySQLadmin and it will be compiled using the MySQL command. PHP code will be written to connect it to the database and it can add questions and answers to the database. By writing HTML we will redevelop the PHP code and MySQL code to be proportional and in the right order so that users can easily read the question, answer them and they not be lost in the page.

Finally with the JqPlot library we will build the chart for the answered questions.

2. PROGRAM OVERVIEW

Creating polls on mobile devices requires using multiple programs and web applications. AJAX Asynchronous JavaScript combined with HTTP and CSS style was used to write a code for this application. DOM was also used because it is a language independent of interacting with objects in XML and HTML. SQL will be used for creating questions and storing the answers in the database.

PHP will be used to create a dynamic web page that can be easily read and understood by a normal user of the Internet.



```

-- DEMO_DES.SQL
--
-- This test generates a table containing a list of names of varying lengths.
-- It then demonstrates how to use the dbo.fn_encrypt_des and
-- dbo.fn_decrypt_des UDF's to encrypt and decrypt the names. Since the whole
-- point of encryption/decryption is to get back exactly what you put in, there is an
-- additional column that indicates whether the Plain Text and decrypted text are
-- equivalent.
--
-- The second part of the test demonstrates the same thing, but using a list of
-- fake SSN's. Note that all of the SSN's in this list have been invalidated by
-- the U.S. SSA (www.ssa.gov), including the 10 SSN's in the list that begin with
-- '987' which have been designated as SSN's that may be used by the public for
-- general purpose use in movies, TV, advertising, etc. For more information on
-- this, visit the SSA website.
--

```

	Plain_Name	Enc_Name	Dec_Name	Success_Flag
1	ADRIANNE	0x7C80CE12807E5339C30F0441E...	ADRIANNE	Y
2	ANNE	0xA23D0AE09C5CD00E	ANNE	Y
3	ANTONIO	0xB9C40FCE3D525BDA	ANTONIO	Y
4	B	0x62EA0BFFC1BC379F	B	Y
5	BILLY	0xE421FFE3C6C35FED	BILLY	Y
6	BRYAN	0x285652EFA7CC95EF	BRYAN	Y
7	CESAR	0xEEA5C92D6D7BDC9F	CESAR	Y
8	CHRISTOPHER	0xE3C58E553079B8DD62FE34A08...	CHRISTOPHER	Y

	Fake_SSN	Enc_Fake_SSN	Dec_Fake_SSN	Success_Flag
1	002281852	0x66F990306663AEF041CC47537...	002281852	Y
2	042103580	0x8440E5D7BCC09C7497A3B8E51...	042103580	Y
3	062360749	0xA5A1E725BDCA6051F0E8531CC...	062360749	Y

Figure 1. SQL Sample Script [1]

Figure 1 illustrates s an SQL database table with people's information in it.

Table 1 PHP Script [2]

```

<?php
$ch = curl_init($sub_req_url);
$encoded = '';
// include GET as well as POST variables; your needs may vary.
foreach($_GET as $name => $value) {
    $encoded .= urlencode($name).'='.urlencode($value).'&';
}
foreach($_POST as $name => $value) {
    $encoded .= urlencode($name).'='.urlencode($value).'&';
}
// chop off last ampersand
$encoded = substr($encoded, 0, strlen($encoded)-1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $encoded);
curl_setopt($ch, CURLOPT_HEADER, 0);
curl_setopt($ch, CURLOPT_POST, 1);
curl_exec($ch);
curl_close($ch);
?>

```

Table 1 shows/describes the PHP script which can be implemented inside the HTML script code.

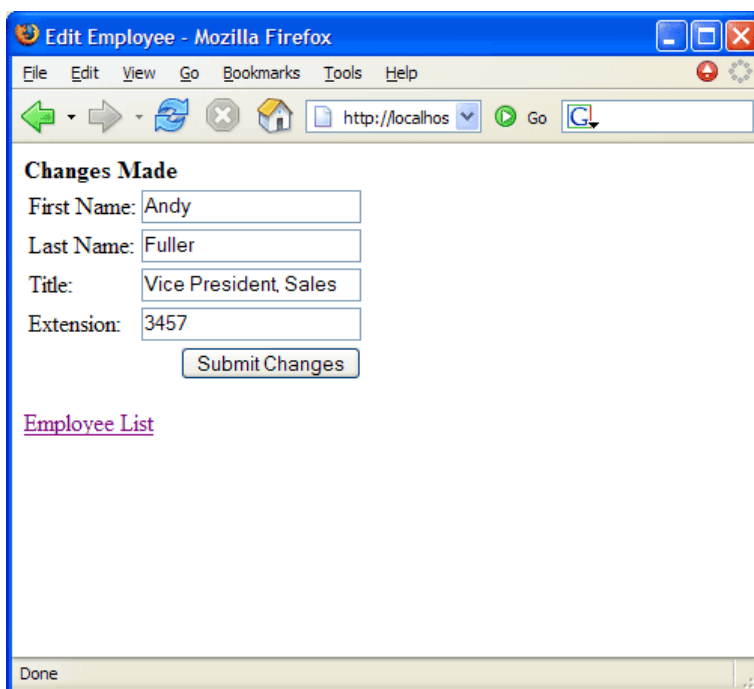


Figure 2. AJAX Sample [3]

Figure 2 represents an AJAX sample with the input fields to make changes and with a button to submit it.

2.1 AJAX: Asynchronous JavaScript

AJAX is an asynchronous JavaScript development method for creating and web applications. AJAX was created to easily send and retrieve data from a server while the visitors are still browsing the web page. It is an application of multiple technologies such as HTML (Hypertext Markup Language) and CSS (*Cascading Style Sheets*). With JavaScript, the DOM (Document Object Mode) is accessed dynamically to display the information presented in the web page.

In early 1990s the web sites were completely built with only HTML. When the user interacts with the web page, the web page is reloaded from the web server or loads another page. So all of the content has to be resent again and again and this places an additional load taking a lot of bandwidth from the web server. In 1995, the JavaScript language was introduced and asynchronous loading was first time tested. This helped the code load asynchronously from the web server after the web site page has been loaded. Microsoft later introduced ActiveX control as the XML HTTP Request JavaScript object which was adopted by other web browsers. In June 2006 the AJAX technology was granted and patented by Microsoft.

The following technologies are incorporated with AJAX:

HTML (Hypertext Markup Language) and CSS (*Cascading Style Sheets*).

DOM (Document Object Mode).

XML (Extensible Markup Language) and XMLHttpRequest.

XSLT (Extensible Stylesheet Language Transformations).

JavaScript combines all these technologies together.

2.2 DOM DOCUMENTED OBJECT MODEL

DOM Documented Object Model is an independent language platform which interacts with HTML and XML. There are 3 DOM levels that were introduced in different years. DOM Level 1 was introduced in early 2000 and it was completely interactive with XML and HTML documents. DOM Level 2 was introduced in late 2000 and started to support the CSS style in the documentation. DOM Level 3 was introduced in 2004 and supported an XPath (XML Path Language) as well as the keyboard handling. In most cases DOM requires a JavaScript to modify a web page. Additionally, HTML objects belong to DOM.

Levels of DOM: Core DOM, HTML DOM, XML DOM.

The DOM Document Object Model is described in Figure 3.

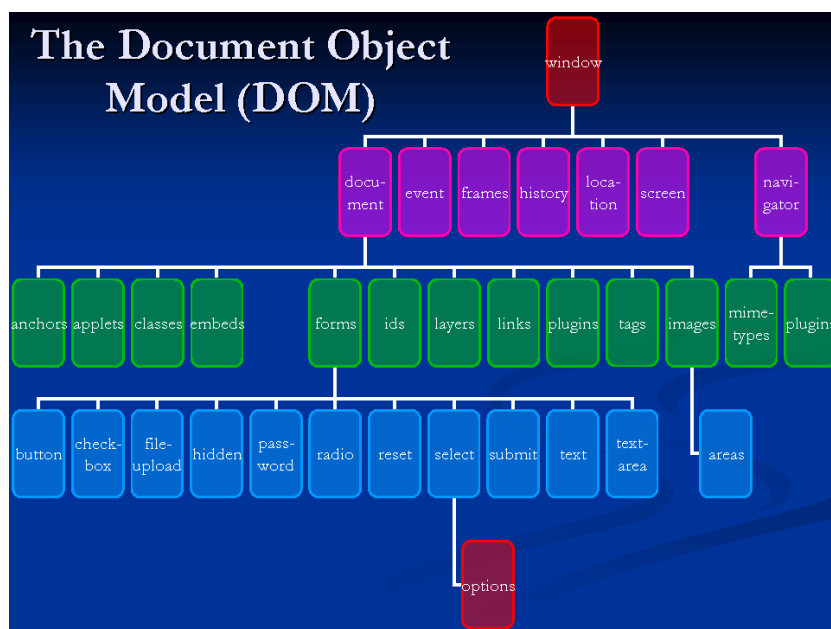


Figure 3. Hierarchy of DOM [4]

2.3 HTTP: Hypertext Transfer Protocol

HTTP Hypertext Transfer Protocol is used by the World Wide Web. It is a protocol written in the URL (Uniform Resource Locator) in the web browser that interacts with the web server that will be directed to the web page that user requested. HTTP is using HTML code to display the environment of the web site that user is browsing. It is case sensitive for mistakes and if a user misspells, it can redirect to another web site. HTTP has thousands of domain names and it is executed independently using technologies such JavaScript, AJAX, ActiveX etc.

It was developed in 1991 with its first version of V0.9 and after 1995 it became more efficient by adding more information using more headers, security protocols and methods.

HTTP is recognized by the written form in URL (Uniform Resource Locator).

`http://www.example.com`

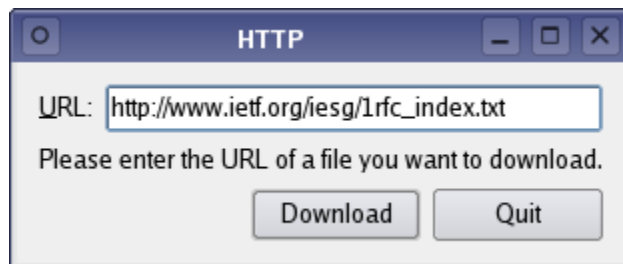


Figure 4. HTTP example [5]

2.4 SQL: Structured Query Language

SQL (Structured Query Language) is a created programming language to manage a database data in a management system based on relational algebra. It is also called a cross-platform operating system. In the SQL database, a user (?) can create, insert, update, and delete data as well as create any types of schema. SQL became the wide used database language worldwide.

Coding in SQL database mostly use this type of queries:

FROM – from where the data can be selected.

WHERE – is a comparison where rows are returned by the query.

ORDER BY – where columns are sorted by a specific order.

GROUP BY – grouping rows and columns when they have the same values.

Here is a simple code for the SQL database:

```
SELECT *  
FROM workers  
WHERE salary > 3000  
ORDER BY firstname;
```

We use this code to find out which workers receive a salary of less than 3000 euro's. The list of workers is ordered according to their first name

2.5 XML: Extensible Markup Language

XML (Extensible Markup Language) is a programming language and a platform to represent data flow in the HTML code. It is a simple code that can be read by any application. It is easy to edit it just like HTML code. It is actually based on the same technology as HTML code. XML is very easy to understand once the basics are learned. XML code is either for programmers or for normal users that can implement it for everyday use in HTML. It can be used in a various different programming languages such as: Java, PHP, C, C++, Perl, Python and others.

Most programmers use XML to create and store data and it can be seen differently on any platform.

In this project XML is used to create data polls questions and to store the answers to the database so that afterwards the results are shown on the graph table. There is a great number of applications where XML is used but it is going to be used in backdrop of HTML coding and together with it so that it will be flexible to find any errors and fix them rather than going through a lot of files.

```
- <xml>
  <title>XML test</title>
  - <text type="test">
    - <body>
      - <p>
        Though this is a very pared
        <br />
down XML document, it nonetheless
        <br />
provides an example of how an XML
        <br />
document displays on the web without
        <br />
the intercession of a stylesheet or
        <br />
other conversion program.
      </p>
    </body>
  </text>
</xml>
```

Table 2. Basic XML table [6]

2.6 JQuery, JQuery Mobile

JQuery is the JavaScript library. It was designed to simplify HTML coding. It is an open source application and any one can use it. With JQuery it is possible to create dynamic web pages very easily.

Ajax, CSS and DOM are contained in the JQuery library which we will be using.

JQuery Mobile is the same library as the JQuery but it is light weight code for the mobile devices such as Black Berry, IPad, IPhone, Nokia, HTC etc. [9] [11]

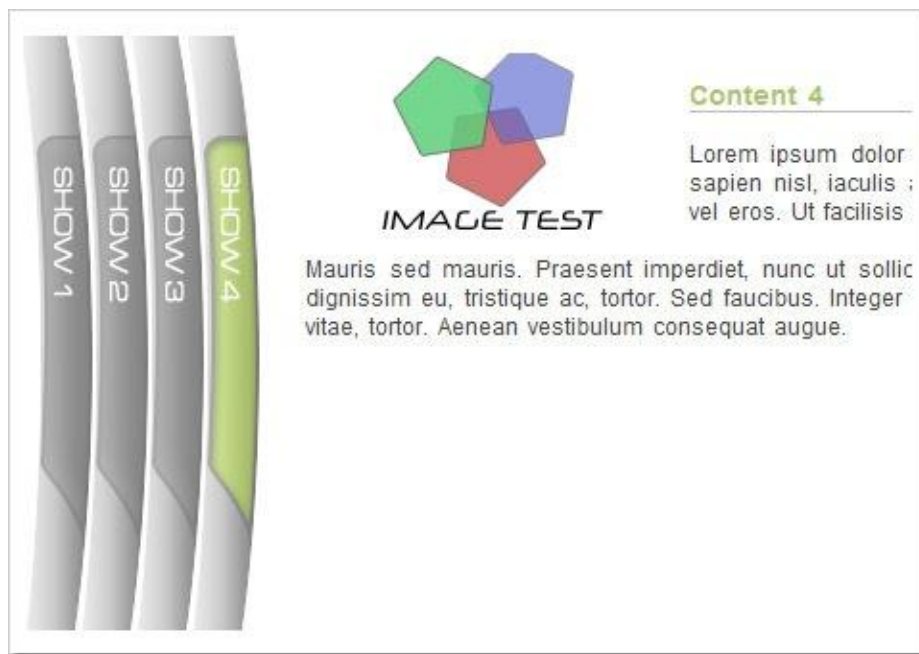


Figure 5. JQuery example [7]

2.7 PHP: Hypertext Preprocessor

PHP is an HTML scripting language and most of its syntax was borrowed from widely known programming languages, such as Perl, C and C++ and Java with of course some specific features of PHP itself. The main goal of the PHP programming language is to allow web programmers to create and develop web pages quickly.

When visitors visit a web site, the server processes the code and then it decides which part is to be shown to visitors and which part is to be hidden from them. For example, text and pictures may be visible to the visitors but the operation of files and calculations will be hidden from them.

All PHP code will be translated by the server into HTML code and will be sent to the visitor's browser for seeing what is on the webpage.

PHP can [13]:

- help to create easy and fast webpages
- help to check thousands of hot scripts online
- Easily create web pages for e-commerce

PHP is a great and powerful script language and by knowing HTML script and some programming language code it is easy to learn it.

```
<table>
<tr>
<td>
  <?php
    echo "php runs first!";
  ?>
</td>
</tr>
</table>
```

the first step is to run php code, then get:

```
<table>
<tr>
<td>
php runs first
</td>
</tr>
</table>
```

then, code is sent to the browser

Table 3: PHP sample table [8]

This is the basic PHP code taken from the PHP website. [13]

3. GENERATING CODE FOR POLLS

3.1 Creating data code for MySQL

Appendix 1.1 displays the MySQL code for creating tables for the database and the code syntax for creating two tables for the MySQL database which will be the main part of the program. The first table displays polls which contain questions. The second table displays the answers for every poll that is there. The second table counts the integer number for every answer that has been voted for.

These two tables are not good enough for the polls. Here there should be another table a voting table that can track those votes so that the visitors should vote only once.

When using MySQL, it is important to build the database first and then run it using the MySQL command.

Here is the code for running RECORD 1: database.sql which can be found in Appendix 1.1.

```
% mysqladmin --user=root --password=foo create quespolls  
% mysql --user=root --password=foo quespolls < database.sql
```

The next step is to create PHP pages when the database has already been created because the database is very important so that when the questions are being created, it will be placed on the database tables which can then be used by the users to answer them. [16]

3.2 Creating PHP pages

The first syntax code represents the questions and answers that will be added into database when it has been created. Appendix 1.2 represents the code for storing answers into the database when they have been created in the HTML code which is displayed in **Figure 6**.

Table x.

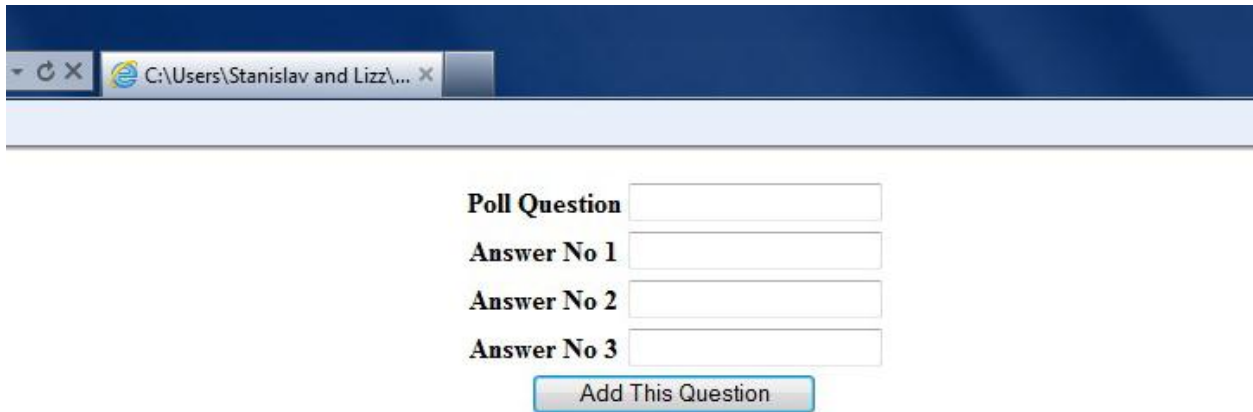
```
<?php
function add_ans ( $db, $qid, $ans ) {
    $sql = 'INSERT INTO ansers VALUES ( 0, ?, ?, 0 )';
    $sth = $db->prepare($sql);
    $sth->execute( array( $qid, $ans ) );
}
```

At first PHP connects the code to the database and checks if the question is actually created. When it has been created, then the new record is created with its unique id and .goes into the polls. After that, the answers are added into the polls question by the add_ans id. Each answer which was created by the add_ans has its own unique id and is added to its unique poll question.

HTML syntax code displays the question in the webpage and the submit button is for the visitors when they are ready to submit the question. [18]

The code in the development mode is displayed in Appendix 1.3.

Figure 6 shows how the HTML table will look like in the web browser. The first field represents the question type and the next three are the answers. At the end, there is an “Add This Question” button for implementing it into the database.



The image shows a screenshot of a web browser window. The address bar displays the path "C:\Users\Stanislav and Lizz\...". The main content area contains a form with the following elements:

- Poll Question**
- Answer No 1**
- Answer No 2**
- Answer No 3**
-

Figure 6. Adding questions

HTML code is written straight after the PHP code. The users can create polling questions by themselves and the other users can answer them.

3.3 Creating the answers PHP page

Creating a PHP answers page first requires to use a connection to the SQL database and secondly to create specific answers for the specific polls. Here we use DOM coding to create an output tag for each answer and for each poll. All the answers have a unique identification so that the answer PHP coding will not be messed. [16]

The code for the PHP page answers can be found in Appendix 1.4.

When the polling questions are answered, the PHP code is connected straight to the MySQL database which stores the answered questions and then JqPlot generates the doughnut graph so that the users can see the results of the poll.

3.4 Generating an answers PHP page

After creating PHP coding function, it should be connected with the answers PHP code.

The count represents on how many visitors had selected this particular answer. For example, answer number one displays “count 10” which means that ten visitors have selected this answer. [16]

3.5 The voting PHP page

After having created and generated answers and answers PHP pages which connect to the SQL database, it is compulsory to create an **Ajax** page so that visitors can easily vote for each poll.

The code for voting PHP page can be found in Appendix 1.6.

There is a difference between the voting PHP page code and the generating answers PHP page code. The voting PHP code is incrementing each answer by one and stores it in the SQL database. [16]

3.6 The polls PHP page

The Script code firstly connects to the MySQL database and then runs the `sql = SELECT` script of the polls table. There are two attributes, the first attribute represents the “id” where each poll tag has unique identification and the second attribute is called “quesn” which represents the unique question tag. Appendix 1.8 shows that a new DOM XML document is created which adds polls and then adds an individual poll. At the end of the code, the document will be saved as XML.

3.7 Creating the front

The web pages are different from pages that have been created for mobile devices. Some of the features are removed for a more optimized version and of course the buttons are larger than in a normal web page on the computer. JQuery Mobile is the source that will help to create web pages for mobile devices with ease.

The code is created with three pages and will be applied within one web HTML page. The first part will show the questions part and the user will select the question that will then show the next part which contains answers for that question. Once the user has selected his/her answer, then, the third part will be shown which contains a chart result for that particular poll.

It is better to write all this code on three different pages because it will be more understanding on how this will work and it is easier to compress it and put it in one HTML web page. The start of the web page is the body which represents the links for the JQuery and JavaScript language.

Second part the “ul” tag is accumulated by the Ajax which will run when the document is compiled. After the document has been compiled, then it contacts the polls PHP page where the XML will return the script from it and then it turns into the “li” tags which are then converted into “ul” tags.

At the home page, the user first clicks the Poll listing and it will return with the answers web page. After the user has chosen the answer for the particular question that he/she chose from the question Poll page, the voting PHP page will be in use after JQuery will plot the doughnut graph to see the results of this poll. Figures 7 – 10 show the screen views of the mobile application.

Let us see Figures 7 to 10 on how the web page will look like on mobile devices.

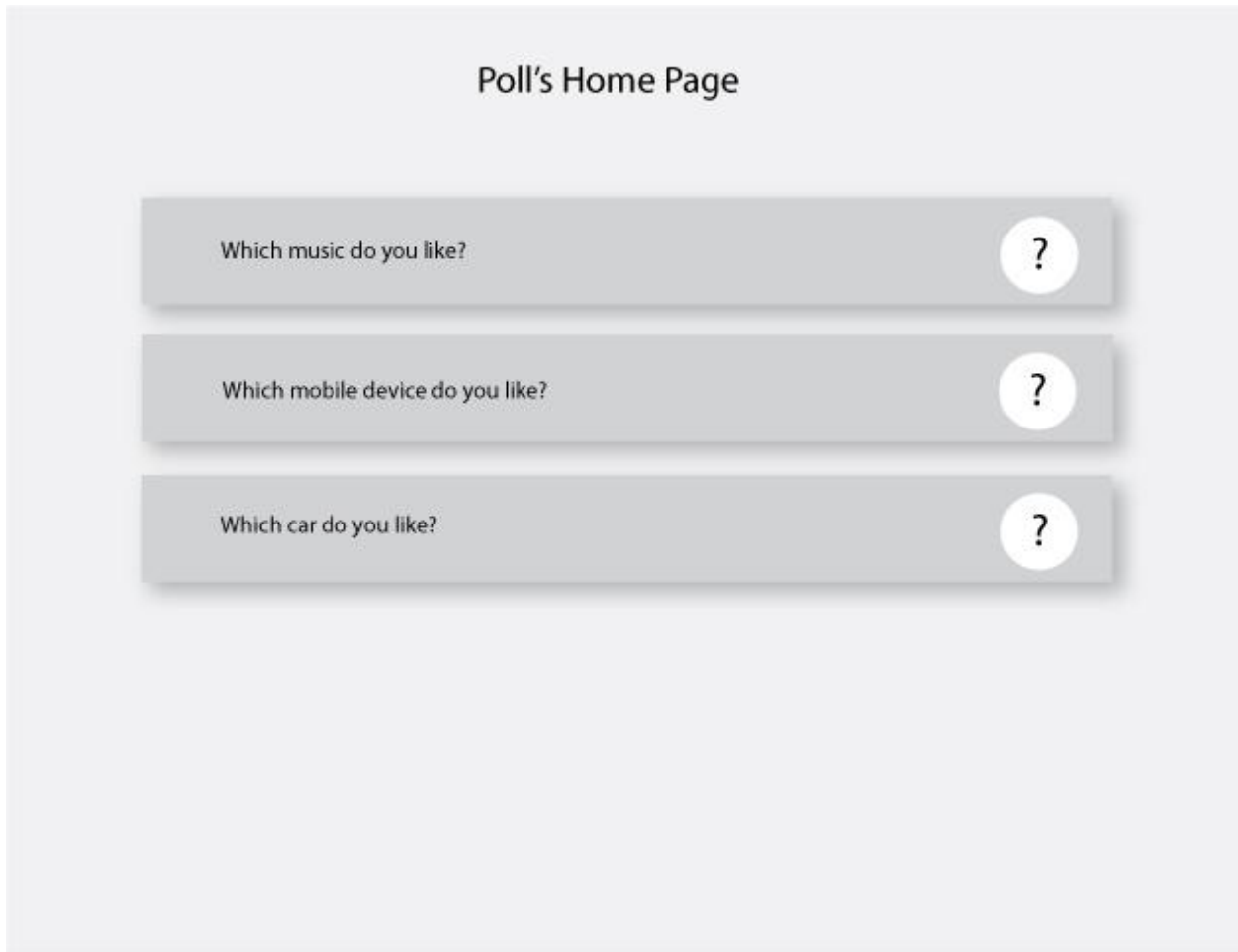
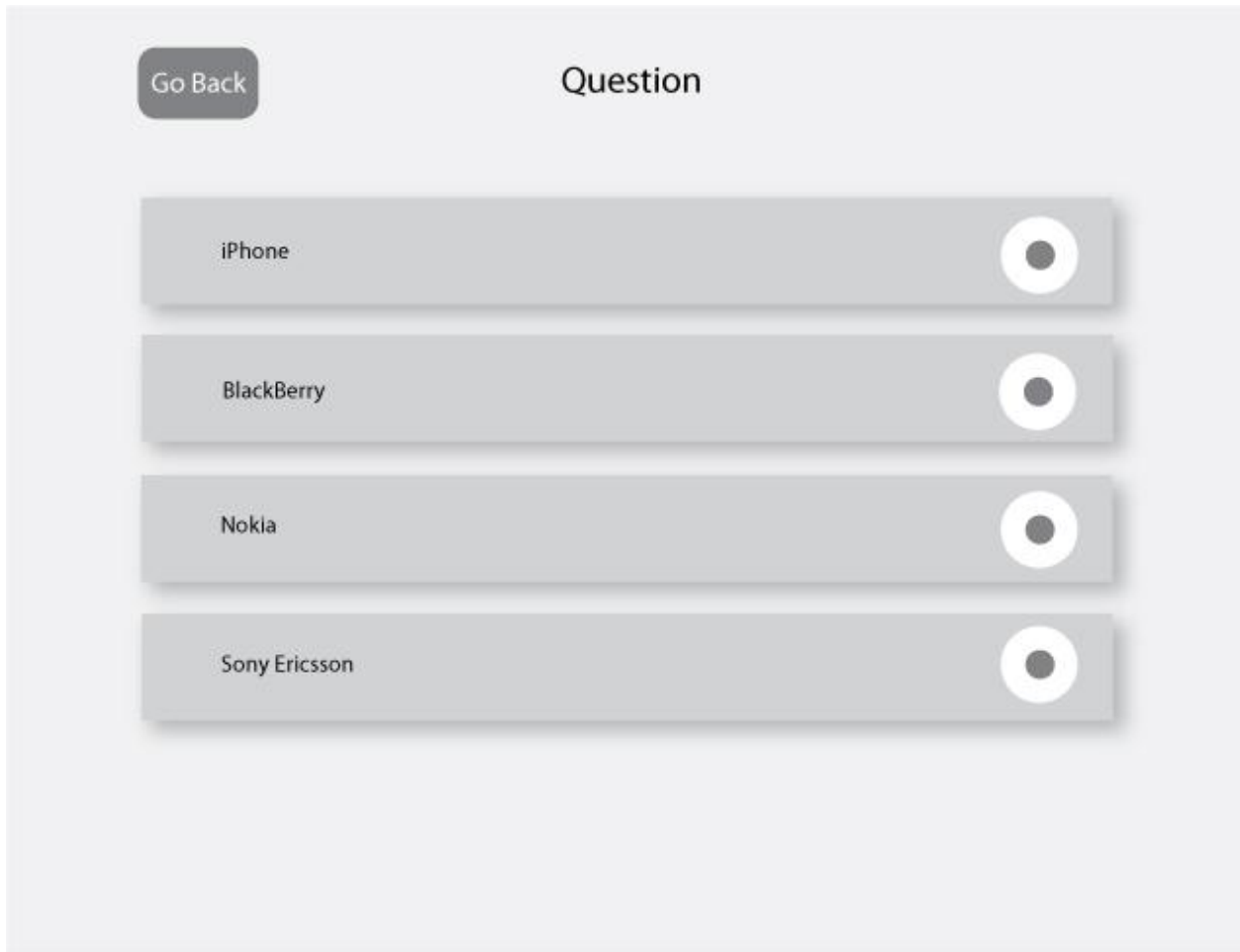


Figure 7: A mobile web page with the Poll's Home Page.

In this figure, there are three types of random questions which can be accessed by clicking a big question mark located on the white circle. After **clicking** the button, it will then show the answers that the user can choose from the answers page which is shown in Figure 11.



The image shows a mobile application interface for a question. At the top left, there is a dark gray button labeled "Go Back". The title "Question" is centered at the top. Below the title, there are four horizontal gray bars, each representing a choice. Each bar contains a device name on the left and a white radio button with a dark gray center on the right. The device names are "iPhone", "BlackBerry", "Nokia", and "Sony Ericsson" from top to bottom.

Figure 8. Answers web page.

Once the user clicks the question mark next to the question “Which mobile device do you like?”, the mobile device screen will display different types of devices that he can **choose.** There is also a “Go Back” button if the user decides not to answer this question.

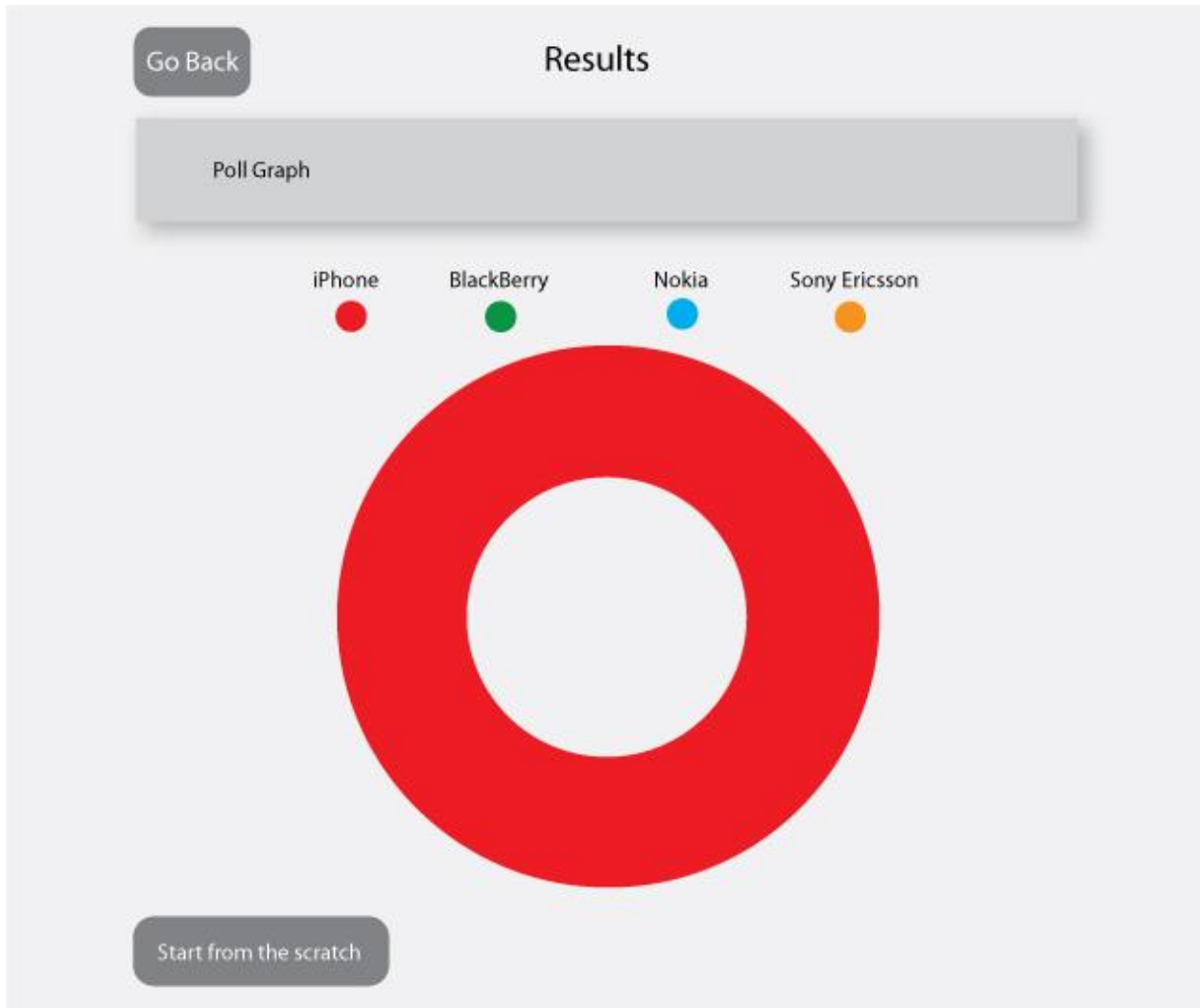


Figure 9: Results web page

Figure 12 shows the final results for the question “Which mobile device do you like?” The result shown in this figure is for the iPhone because the question was only answered by one person. More answers are shown in Figure 10.

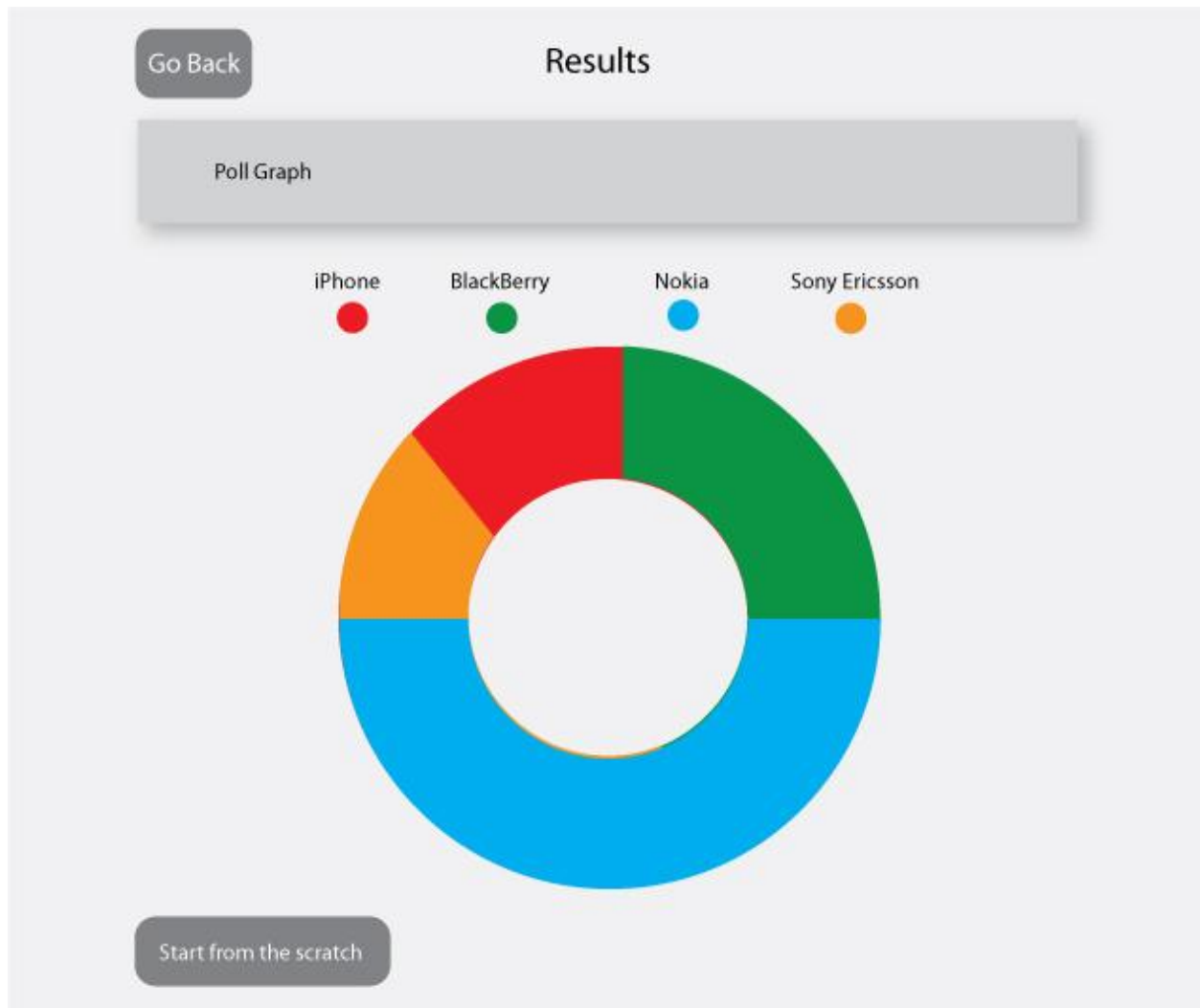


Figure 10. Results with multiple votes

Figure 13 show the results with multiple votes for the question “Which mobile device do you like?”

4 SUMMARY

This thesis is based on the discussions about famous programs and programming languages such as Ajax, JavaScript, PHP, JQuery, jQuery Mobile. These programming languages, from the JavaScript in the early days up to JQuery and JQuery Mobile nowadays for the mobile devices, are widely used in the modern world.

Implementing such web pages like Polls for Mobile Devices can be very useful and more efficient in the modern world. Answering polls on paper and then filing them all together and analyzing all the answers to obtain the graph result can take a long time to do it. With technology such JQuery we simply create code for the device and then users can answer the polls wherever they are because of the wireless technology and access to the Internet. There is no need to gather the answers and then wait for days to obtain the graph result as a user can simply navigate to the polls page from his/her web browser and answer the desired questions.

The result of the simple polling structure is shown at the syntax code. Just by knowing the basics programming languages that been used in this thesis it is simple to create such pages.

This is basic code for making polls but there is more complicated code for more sophisticated polls. Programming codes are always being refreshed and will be improving day by day. However, our life is becoming easier with the Internet rather than carrying bunch of papers that will be recycled after.

References

- [1] SQL Sample script [www-document]. Available at:
http://www.sqlservercentral.com/columnists/mcoles/SAMPLE_SCRIPT1.GIF
(Accessed 18 January 2012)
- [2] PHP Basic Culr example[www-document] Available at:
<http://fi2.php.net/manual/en/curl.examples.php>
(Accessed 19 January 2012)
- [3] Ajax Employees example [www-document] Available at:
<http://www.learn-ajax-tutorial.com/Images/EditEmployee2.gif>
(Accessed 22 January 2012)
- [4] Model DOM table [www-document] Available at:
http://1.bp.blogspot.com/_AU3RuK_Z0g/TmrXLePOp_I/AAAAAAAAAD30/dUNxPPU2v4/s1600/the-document-object-model-dom.gif
(Accessed 24 January 2012)
- [5] HTTP example [www-document] Available at:
<http://doc.qt.nokia.com/4.7-snapshot/images/http-example.png>
(Accessed 28 January 2012)
- [6] XML example [www-document] Available at:
http://cdrh.unl.edu/articles/guide_site/images/xml1.gif
(Accessed 29 January 2012)
- [7] JQuery example [www-document] Available at:
<http://cdn1.1stwebdesigner.com/wp-content/uploads/2011/03/jquery-nav.jpg>
(Accessed 30 January 2012)

- [8] Introduction of PHP [www-document] Available at:
<http://www.php.net/manual/en/introduction.php>
(Accessed 5 February 2012)
- [9] JQuery web site[www-document] Available at:
<http://jquery.com/>
(Accessed 13 February 2012)
- [10] JQplot web site[www-document] Available at:
<http://www.jqplot.com/>
(Accessed 13 February 2012)
- [11] JQuery mobile web site[www-document] Available at:
<http://jquerymobile.com/>
(Accessed 14 February 2012)
- [12] HTML elements[www-document] Available at:
<http://htmlhelp.com/reference/wilbur/overview.html>
(Accessed 19 February 2012)
- [13] PHP web site[www-document] Available at:
php.net
(Accessed 30 February 2012)
- [14] The World Wide Web Consortium web site [www-document] Available at:
<http://www.w3.org/>
(Accessed 19 March 2012)
- [15] PHP introduction[www-document] Available at
<http://php.net/manual/en/introduction.php>
(Accessed 25 March 2012)
- [16] Robin Nixon - (2009) Learning PHP, MySQL, and JavaScript. Place of Publication:
O'Reilly Media

[17] HTML web site[www-document] Available at <http://htmlbook.ru/>
(Accessed 19 January 2012)

[18] Andy Harris - John Wiley & Sons (2010)

HTML, XHTML and CSS All-In-One For Dummies Place of Publication: John Wiley & Sons

Appendix 1.1

RECORD 1: database

Table 1: Represent questions for visitors

```
DROP TABLE IF EXISTS quespolls;  
CREATE TABLE quespolls(  
  id INT NOT NULL AUTO_INCREMENT,  
  ques TEXT NOT NULL,  
  primary key ( id ) );
```

Table 2. Represent answers for visitors

```
DROP TABLE IF EXISTS anspolls;  
CREATE TABLE anspolls(  
  id INT NOT NULL AUTO_INCREMENT,  
  poll INT NOT NULL,  
  ans TEXT NOT NULL,  
  count INT,  
  primary key ( id ) );
```

Appendix 1.2

```

$dd = new PDO('mysql:host=localhost;dbname=anspolls', 'root', '');
if ( isset( $_POST['quesn'] ) ) {
    $sql = 'INSERT INTO anspolls VALUES ( 0, ? )';
    $sth = $dd->prepare($sql);
    $sth->execute( array( $_POST['quesn'] ) );
    $qid = $dd->lastInsertId();
    if ( isset( $_POST['ans1'] ) && strlen( $_POST['ans1'] ) > 0 )
        add_ans( $dd, $qid, $_POST['ans1'] );
    if ( isset( $_POST['ans2'] ) && strlen( $_POST['ans2'] ) > 0 )
        add_ans( $dd, $qid, $_POST['ans2'] );
    if ( isset( $_POST['ans3'] ) && strlen( $_POST['ans3'] ) > 0 )
        add_ans( $dd, $qid, $_POST['ans3'] );
    if ( isset( $_POST['ans4'] ) && strlen( $_POST['ans4'] ) > 0 )
        add_ans( $dd, $qid, $_POST['ans4'] );
    if ( isset( $_POST['ans5'] ) && strlen( $_POST['ans5'] ) > 0 )
        add_ans( $dd, $qid, $_POST['ans5'] );
}
?>

```

Appendix 1.3

```

<html>
<body>
<form method="post">
<center>
<table>
<tr><th>Poll Question</th><td><input type="text" name="quesn" /></td></tr>
<tr><th>Answer No 1</th><td><input type="text" name="ans1" /></td></tr>
<tr><th>Answer No 2</th><td><input type="text" name="ans2" /></td></tr>
<tr><th>Answer No 3</th><td><input type="text" name="ans3" /></td></tr>
</table>
<input type="submit" value="Add This Question" />
</center>
</form>
</body>
</html>

```

Appendix 1.4

```

<?php

function add_ans ( $db, $qid, $ans ) {
    $sql = 'INSERT INTO ansers VALUES ( 0, ?, ?, 0 )';
    $sth = $db->prepare($sql);
    $sth->execute( array( $qid, $ans ) );
}

$d = new PDO('mysql:host=localhost;dbname=anspolls', 'root', '');
if ( isset( $_POST['quesn'] ) ) {
    $sql = 'INSERT INTO anspolls VALUES ( 0, ? )';
    $sth = $d->prepare($sql);
    $sth->execute( array( $_POST['quesn'] ) );
    $qid = $d->lastInsertId();
    if ( isset( $_POST['ans1'] ) && strlen( $_POST['ans1'] ) > 0 )
        add_ans( $d, $qid, $_POST['ans1'] );
    if ( isset( $_POST['ans2'] ) && strlen( $_POST['ans2'] ) > 0 )
        add_ans( $d, $qid, $_POST['ans2'] );
    if ( isset( $_POST['ans3'] ) && strlen( $_POST['ans3'] ) > 0 )
        add_ans( $d, $qid, $_POST['ans3'] );
    if ( isset( $_POST['ans4'] ) && strlen( $_POST['ans4'] ) > 0 )
        add_ans( $d, $qid, $_POST['ans4'] );
    if ( isset( $_POST['ans5'] ) && strlen( $_POST['ans5'] ) > 0 )
        add_ans( $d, $qid, $_POST['ans5'] );
}
?>
<html>
<body>
<form method="post">
<center>
<table>
<tr><th>Poll Question</th><td><input type="text" name="quesn" /></td></tr>
<tr><th>Answer No 1</th><td><input type="text" name="ans1" /></td></tr>
<tr><th>Answer No 2</th><td><input type="text" name="ans2" /></td></tr>
<tr><th>Answer No 3</th><td><input type="text" name="ans3" /></td></tr>
</table>
<input type="submit" value="Add This Question" />
</center>
</form>
</body>
</html>

```


Appendix 1.5

```
<?php
```

```
function create_ansers( $dbh, $poll )
{
    $sql = 'SELECT * FROM ansers where poll=?';

    $q = $dbh->prepare ( $sql );
    $q-> execute ( array( $poll ) );

    $doc = new DOM Document();
    $r = $doc-> createElement ( "ansers" );
    $doc->appendChild( $r );

    foreach ( $q->fetchAll() as $row)
    {
        $e = $doc->createElement( "ans" );
        $e->setAttribute( 'id', $row['id'] );
        $e->setAttribute( 'answer', $row['ans'] );
        $e->setAttribute( 'count', $row['count'] );
        $r->appendChild( $e );
    }

    print $doc->saveXML();
}

?>
```

Appendix 1.6

```
<?php
require_once( 'build_ansers.php' );

header( 'Content-Type:text/xml' );

$dbh = new PDO('mysql:host=localhost;dbname= anspolls ', 'root', '');
create_ansers( $dbh, $_REQUEST['id'] );

?>
```

This code connects to SQL database and sends the identification to the Creating answers PHP code.

Here is another code of the answers examples for poll question.

```
$ curl "http://localhost/poll/ansrs.php?id=1"

<?xml version="1.0"?>
<ansers>
  <answer no 1 id="1" answer="Its Fantastic" count="10"/>
  <answer no 2 id="2" answer="It's good" count="7"/>
  <answer no 3 id="3" answer="It's aright" count="2"/>
  <answer no 4 id="4" answer="Pretty bad" count="4"/>
</ansers>

$
```

Appendix 1.7

```
<?php

require_once( 'create_ansers.php' );

header('Content-Type: text/xml');

$poll = 0;

$dd = new PDO('mysql:host=localhost;dbname= anspolls ', 'root', '');
if ( isset( $_REQUEST['id'] ) )
{
    $sth = $dd->prepare("SELECT count, poll FROM ansers WHERE id=?");
    $sth->execute( array( $_REQUEST['id'] ) );
    $count = 0;
    foreach ( $sth->fetchAll() as $row )
    {
        $count = $row['count'];
        $poll = $row['poll'];
    }
    $count++;
    $sql = 'UPDATE ansers SET count=? WHERE id=?';
    $sth = $dd->prepare($sql);
    $sth->execute( array( $count, $_REQUEST['id'] ) );
}

create_ansers ( $dd, $poll );

?>
```

Appendix 1.8

```
<?php
Header ( 'Content-Type: text/xml' );

$dbh = new PDO ('mysql:host=localhost; dbname=anspolls', 'root', '');

$sql = 'SELECT * FROM polls';

$q = $dbh->prepare( $sql );
$q-> execute( array() );

$doc = new DocumentDOMXML ();
$r = $doc->createElement( "polls" );
$doc-> appendChild( $r );

foreach ( $q->fetchAll() as $row)
{
    $e = $doc->createElement( "poll" );
    $e-> setAttribute( 'id', $rows['id'] );
    $e-> setAttribute( 'quesn', $row['quesn'] );
    $r-> appendChild( $e );
}

print $doc->saveXML();

?>
```

Appendix 1.9

First part:

```
<html>
<head>
<link rel="stylesheet" href="css/jquery.mobile-1.0a4.1.css" />
<link rel="stylesheet" type="text/css" href="css/jquery.jqplot.css" />
<script src="js/jquery-1.6.1.min.js">
</script>
<script src="js/jquery.mobile-1.0a4.1.js">
</script>
<script language="javascript" type="text/javascript" src="js/jquery.jqplot.js">
</script>
<script language="javascript" type="text/javascript"
src="js/plugins/jqplot.donutRenderer.js">
</script>
```

Second part:

```
<script type="text/javascript">
function plotData( data )
{
ds = [];
$(data).find('ans').each( function()
{
ds.push( [ $(this).attr('ans'), parseInt( $(this).attr('count') ) ] ); } );
$.jqplot('graph1', [ds],
{
```

```

seriesDefaults:
{
  renderer:$.jqplot.DonutRenderer
  },
  legend: {show:true} });
}
function vote( poll, ans )
{
$.ajax( { url: 'voting.php',
data:{id:ans},
success:function( data )
{
plotData( data );
}
});
}
function openpoll( poll )
{
$.ajax( { url: 'ansers.php',
data:{id:poll},
success:function( data ) {
$(data).find('ans').each( function()
{
var name = $(this).attr('ans');
var id = $(this).attr('id');
$('#ans-list').append( '<li><a href="#results" poll="'+poll+'" ans="'+id+"'
class="ans">'+name+'</a></li>' ); } );
$('.answer').click(function(e)
{
vote( $(this).attr('poll'),
$(this).attr('ans') );})
$('ul').listview('refresh');
}
});
}
}

```

```
$(document).ready(function()
{
$.jqplot.config.enablePlugins = true;
$.ajax( { url: 'polls.php',
success:function( data )
{

$(data).find('poll').each( function()
{
var name = $(this).attr('quesn');
var id = $(this).attr('id');
$('#poll-list').append(
'<li><a href="#answers" pollid="'+id+'" class="poll">'+name+'</a></li>' ); });
$('.poll').click(function(e)
{
openPoll( $(this).attr('pollid') ); })
$('.ul').listview('refresh'); } });});

</script>
```

Third part:

```

</head>
<body>
<div data-role="page" id="home">
<div data-role="header">
<h1>
Poll's Home Page
</h1>
</div>
<div data-role="content">
<ul data-role="listview" data-inset="true" data-theme="c" data-dividertHEME="b"
id="poll-list">
</ul>
<div id="chart0" class="plot" style="width:350px;height:200px;">
</div>
</div>
</div>

<div data-role="page" id="ans">
<div data-role="header">
<h1 id="poll-quesn">
Question
</h1>
</div>
<div data-role="content">
<ul data-role="listview" data-inset="true" data-theme="c" data-dividertHEME="b"
id="answer-list">
</ul>
</div>
</div>

```



```
<div data-role="pageresults" id="results">
<div data-role="header">
<h1>
Results
</h1>
</div>
<div data-role="content">
<div data-role="collapsible">
<h3>
Poll Graph
</h3>
<p>
<div class="jqPlot" id="chart1" style="height:300px; width:440px;">
</div>
</p>
</div>

<a href="#home" data-role="button">Start from the scratch</a>
</div>
</div>
</body>
</html>
```

Appendix 1.10

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<meta name="viewport" content="width=device-width, initial-scale=1" />
<title>
</title>
<link rel="stylesheet" href="http://code.jquery.com/mobile/1.1.0/jquery.mobile-1.1.0.min.css" />
<style>
/* App custom styles */
</style>
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.js">
</script>
<script src="http://code.jquery.com/mobile/1.1.0/jquery.mobile-1.1.0.min.js">
</script>
</head>
<body>
<div data-role="page" id="page1">
<div data-theme="" data-role="header">
<h5>
Poll's Home Page
</h5>
```

```
</div>
<div data-role="content">
<a data-role="button" data-transition="fade" data-theme="b" href="#page1">
  Which music do you like?
</a>
<a data-role="button" data-transition="fade" data-theme="b" href="#page1">
  Which mobile device do you like?
</a>
<a data-role="button" data-transition="fade" data-theme="b" href="#page1">
  Which car do you like?
</a>
</div>
</div>
<script>
//App custom javascript
</script>
</body>
</html>
```

Testing the JQuery Mobile application

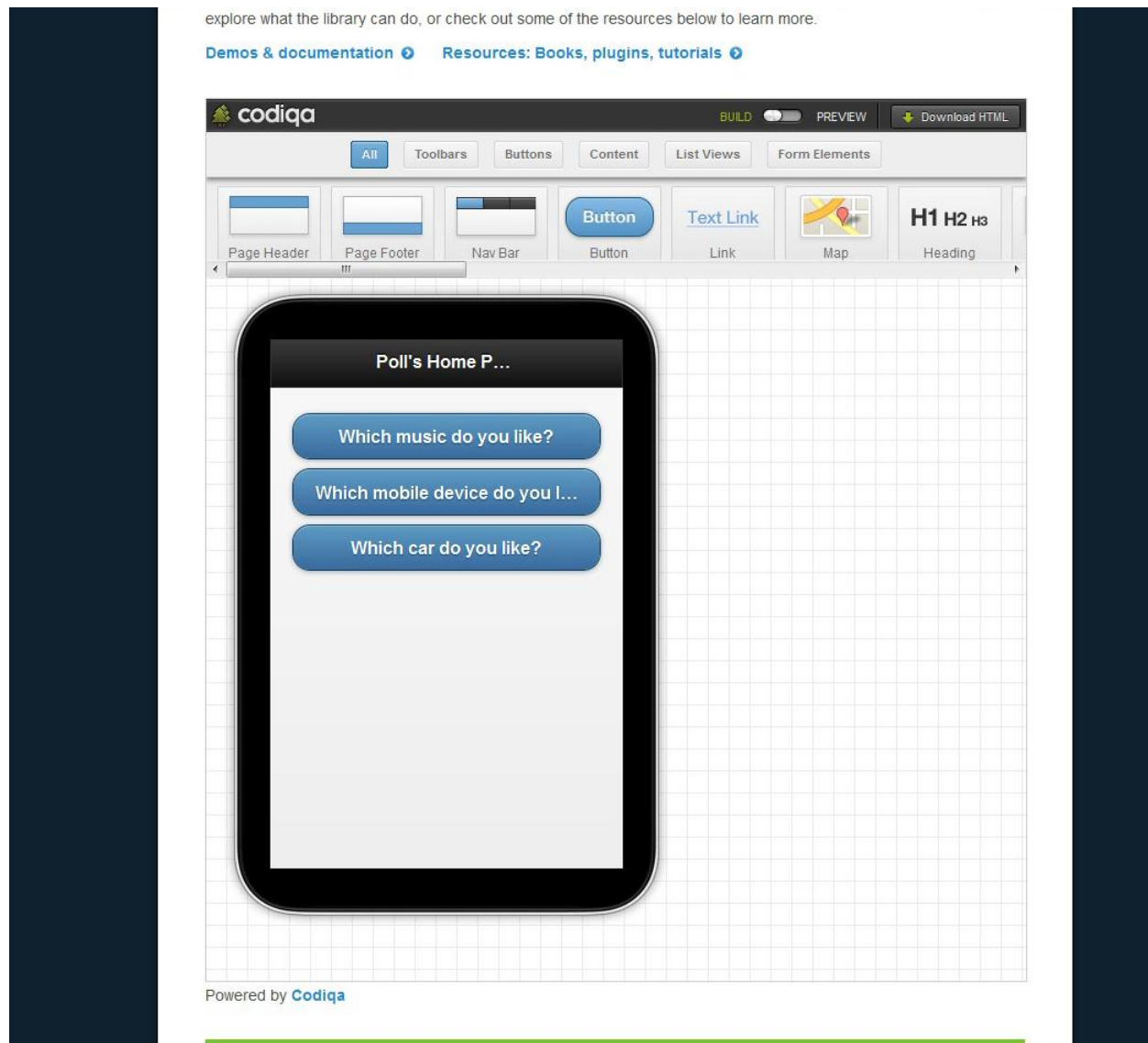


Figure 11: JQuery web page for mobile device

Figure 14 shows the actual test web page for mobile device such as iPad. This is the actual look of the Poll's Home Page where the questions is been presented. It is very simple just go to www.jquerymobile.com and start creating a small poll.

Once the creation of the web page for a mobile device is finished, it is then simple to download HTML code from the web site. In Figure 14 there is a button in the right top corner where it says Download HTML, simply click on it and it will give you the HTML code that have been created.

The HTML code can be found in Appendix 1.10.