A WEB APPLICATION FOR ONLINE POLLING

_______________

A Thesis
Presented to the
Faculty of
San Diego State University

_______________

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
in
Computer Science

_______________

by
Ashok Ramachandra
Spring 2015
SAN DIEGO STATE UNIVERSITY

The Undersigned Faculty Committee Approves the

Thesis of Ashok Ramachandra:

A Web Application for Online Polling

__________________________
Carl Eckberg, Chair
Department of Computer Science

__________________________
Wei Wang
Department of Computer Science

__________________________
Janet Bowers
Department of Mathematics and Statistics

April 8, 2015
Approval Date
DEDICATION

I would like to dedicate this work of Thesis to my family, and friends for their continuous support. I express my gratitude to them.
ABSTRACT OF THE THESIS

A Web Application for Online Polling
by
Ashok Ramachandra
Master of Science in Computer Science
San Diego State University, 2015

It can sometimes be tricky for the companies to know what the users actually feel about their product. Feedback from the customers is very important.

This thesis focuses on creating a Web application for Online Pooling. People from remote locations having different opinions can give their feedback in the form of a vote. People can select products from a wide range of categories and cast their vote. This application has an interactive User Experience which makes the software easy to use. By making this application online, users can cast their vote from any Internet enabled computer or handheld device. This application makes use of responsive design so that it can be used on Computers, Tablets and Mobile Devices.
TABLE OF CONTENTS

PAGE

ABSTRACT ............................................................................................................................... v
LIST OF TABLES .................................................................................................................. viii
LIST OF FIGURES ............................................................................................................... ix
LIST OF ABBREVIATIONS ............................................................................................... x
ACKNOWLEDGEMENTS ................................................................................................... xi

CHAPTER

1 INTRODUCTION .................................................................................................................. 1

2 APPLICATION DESIGN ................................................................................................. 3
   2.1 Overview .................................................................................................................. 3
   2.2 Hardware Requirements ....................................................................................... 3
   2.3 Software Requirements ....................................................................................... 3
   2.4 Architecture .......................................................................................................... 4
   2.5 Application Overview ........................................................................................... 5

3 IMPLEMENTATION ......................................................................................................... 7
   3.1 Style Sheet ............................................................................................................. 7
      3.1.1 Login ............................................................................................................. 7
      3.1.2 Registration ................................................................................................. 9
      3.1.3 Create Topic ............................................................................................... 11
      3.1.4 Cast Vote ................................................................................................... 14
   3.2 Features .................................................................................................................. 16
      3.2.1 Register View .............................................................................................. 16
      3.2.2 Log-in View ............................................................................................... 17
      3.2.3 Create Topic ............................................................................................... 18
      3.2.4 Voting ......................................................................................................... 19

4 TESTING ........................................................................................................................... 21
4.1 Functionality Testing .................................................................21
4.2 Security Testing .........................................................................22
5  SUMMARY ...................................................................................23
   5.1 Conclusion ...............................................................................23
   5.2 Obstacles Faced .....................................................................23
6  FUTURE EVOLUTION ....................................................................24
REFERENCES ..................................................................................25
LIST OF TABLES

Table 2.1: Hardware Requirements .......................................................... 3
Table 2.2: Software Requirements ........................................................... 4
LIST OF FIGURES

Figure 2.1: Technology stack.................................................................4
Figure 2.2: Working diagram...............................................................6
Figure 3.1: Login template.................................................................8
Figure 3.2: Signup template...............................................................11
Figure 3.3: Create Topic template..................................................13
Figure 3.4: Cast Vote template.........................................................16
Figure 3.5: Register.................................................................17
Figure 3.6: Login.................................................................18
Figure 3.7: Create Topic...............................................................19
Figure 3.8: Voting.................................................................20
Figure 4.1: Database connectivity testing........................................21
Figure 4.2: Security testing...........................................................22
### LIST OF ABBREVIATIONS

1. HTML  
   Hyper Text Markup Language
2. CSS  
   Cascading Style Sheet
3. SQL  
   Structured Query Language
4. MAMP  
   Mac apache MySQL and PHP
5. API  
   Application Programming Interface
6. UI  
   User Interface
ACKNOWLEDGEMENTS

I thank Dr. Carl Eckberg from the Computer Science department for being the Chair of the Thesis committee and for his advice throughout my time on this Thesis. I thank Dr Wei Wang from the Computer Science department for serving as a member in my committee. I would also like to convey my regards to Dr. Janet Bowers from the Department of Mathematics and Statistics for serving as the third member in my Thesis committee.)
CHAPTER 1

INTRODUCTION

Since its inception, the internet has offered new methods of conducting research. One such way is the use of web-based surveys. Up until the last several years, the primary methods of survey research included telephone, mail, and face to face interviewing. Now, with the internet, it is possible to conduct web-based surveys in addition to traditional methods.

Due to the immense competition in the world, it has become the most important task, to provide better solutions to complex problems. In order to be near optimal solution, there should be a high level of efficiency, effectiveness, and feasibility. These types of solutions are obtained through the combined efforts of multiple people with different viewpoints. Collaboration is of paramount importance since it helps to increase innovation and enhance efficiency.

There have been many instances where several viewpoints or opinions have been expressed from remote locations have helped in arriving at a solution for problems. But before that, the priorities and concerns of the Requirement Owner needs to be considered. To obtain dynamic data of this kind, proactive collaboration between the Solution Provider and the Requirement Owner is mandatory. There is a great need for an online voting application, which is the topic of this thesis, which can get fast feedback from remote collaborators, or an intended user group.

There are some voting tools available online. Few of them are:

- Easypolls, A polling tool created by Object Planet Inc. This application makes use of third party software like Fabrastic, Modernizr. It is mostly written in Javascript [1].
- SurveyMonkey, which is an online survey tool. SurveyMonkey provides customizable surveys and paid backend programs [2].
• Poll Everywhere, an online polling web and mobile application, which works on cloud lets users engage from anywhere in real time [3].

The Web Application here is also a rank order voting tool meaning that the votes are cast by users in the order of their priority.

The Report is divided into five chapters:
• Chapter 1: Introduction to this Online Polling Web Application.
• Chapter 2: Describes the architecture of the software and the technology used.
• Chapter 3: Mentions how the application was implemented
• Chapter 4: Explains how testing was done
• Chapter 5: Summary along with some of the major problems faced
• Chapter 6: Future evolution is discussed
CHAPTER 2

APPLICATION DESIGN

2.1 OVERVIEW

A Web Application for online polling allows users to create ballot lists and use these ballot items to vote with the help of a Graphical User Interface. The main focus of this document is the front end and middleware of the polling application. The application is built using MAMP (Mac, Apache, MySQL and PHP). The client side of this application communicates with the database with the help of middleware. The results are then displayed on the front end. The front end is built using HTML, CSS and JavaScript.

In order to make this web application available for users in different locations, we need to make this application online. For that we need

- Domain Name
- Web hosting

2.2 HARDWARE REQUIREMENTS

These are the hardware requirements for online polling web application

<table>
<thead>
<tr>
<th>Table 2.1: Hardware Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor</td>
</tr>
<tr>
<td>RAM</td>
</tr>
<tr>
<td>Hard Disk Drive</td>
</tr>
</tbody>
</table>

2.3 SOFTWARE REQUIREMENTS

These are the software requirements for online polling web application
Table 2.2: Software Requirements

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating System</td>
<td>MAC OSX 10.6 and above</td>
</tr>
<tr>
<td>Web Browser</td>
<td>Firefox, Chrome</td>
</tr>
<tr>
<td>IDE</td>
<td>Sublime text</td>
</tr>
</tbody>
</table>

2.4 Architecture

For this application we use MAMP platform. MAMP is composed of open source and proprietary software to run websites dynamically on Apple computers.

![MAMP Architecture](image)

**MAMP Architecture**

- **Mac - OS**
- **Apache - Web**
- **MySQL - DB**
- **PHP - Script**

**Figure 2.1: Technology stack.**

- **Browser**
  - A web browser is a software application for retrieving, presenting and traversing information resources on the World Wide Web.
  - We can use any browser for this tool as long as it supports responsive design.
• **Operating System**
  • An operating system (OS) is software that manages computer hardware and software and provides common services for computer programs.
  • We use Mac OS for this Application.

• **Web Service**
  • A Web service is a method of communication between two electronic devices over a network
  • We use Apache Web service for this application [4].

• **Database**
  • A Relational Database System is used on the web to store and edit data
  • We use My SQL Database for this application [5].

• **PHP**
  • A server side scripting language which can be embedded into HTML
  • PHP can used to generate an HTML code of a page. It can also generate an image.

• **HTML5 (Hyper Text Markup Language)**
  • Used to design a Web page
  • Helps us communicate with the others on the World Wide Web

• **CSS3 (Cascading Style Sheets)**
  • Used to define the visual styles of the web pages
  • Separates design form the content

• **JavaScript**
  • Used to add interactions to web page
  • Embed to HTML code

• **Flask**
  • Flask is a framework written in Python.
  • Flask does not fore the developers to use a particular tool or library.

2.5 **APPLICATION OVERVIEW**

This application enables the admin to create, edit or delete polls. Admin can delete a user from the database. Users and admin can both see the results.
The suggested tool will enable the users to articulate their views and provide feedback in order to establish a path for the design team to work.

The tool consists of:

- Ballot
- Administrators (Create Ballot)
- Users (Caste Vote)
- Ballot items (A ballot can have multiple ballot items)
- Vote

This is an online application, so the users can cast their vote from any Internet enabled computer or handheld device.
CHAPTER 3

IMPLEMENTATION

3.1 STYLE SHEET

Style sheets are used to separate presentation and content in a web application. While the HTML contains the structure, CSS defines the visual layout.

There are various benefits in using style sheets

- **SPEED**
  Sites which use style sheets are generally quicker than those which don’t. This is because the CSS files will be present in the Browser’s cache.

- **CONSISTENCY**
  In an application it is important to maintain consistency. Using style sheets, the presentational details can be specified and a single style sheet can be used for several web pages, saving repetitive code and errors.

- **MAINTAINABILITY**
  It is easy to maintain the style guides if we maintain one presentation file. This reduces the maintenance time and the risk of errors.

3.1.1 Login

In order to use this web application, the user needs to be authenticated. It is necessary to keep trespassers away from some of the confidential information present in the web application. Once the user enters all the necessary information to login, and clicks the login button, HTML connects with the middleware and checks if the user is authenticated. If the user id and the password do not match, then an error message is displayed and the user id and password field are set to blank [6]. In order to login successfully, users need to be registered and have to enter correct login details. Figure 3.1 shows the template of the login page.
Figure 3.1: Login template.
3.1.2 Registration

To vote, a user has to register in the site with his valid information. After the registration, the user can login into the site and give his opinion. To register, the user has to click Signup link, which takes the user to the register page. The figure below shows the template of the SignUp page.

```html
<form id = "formSignUp" class="form-horizontal" action="signup.html" method = "post" >
    <div class="form-group">
        <!-- <label for="inputEmail3" class="col-sm-2 control-label">Email</label> -->
        <div class="col-sm-10">
            <input type="text" class="form-control" id="inputEmail3" placeholder="Name" name="name">
        </div>
    </div>
    <div class="form-group">
        <!-- <label for="inputEmail3" class="col-sm-2 control-label">Email</label> -->
        <div class="col-sm-10">
            <input type="text" class="form-control" id="inputEmail3" placeholder="Username" name="username">
        </div>
    </div>
    <div class="form-group">
        <!-- <label for="inputEmail3" class="col-sm-2 control-label">Email</label> -->
        <div class="col-sm-10">
            <input type="email" class="form-control" id="inputEmail3" placeholder="Email" name="email">
        </div>
    </div>
</form>
```
<!- <label for="inputEmail3" class="col-sm-2 control-label">Email</label> -->
<div class="col-sm-10">
<input type="text" class="form-control" id="inputEmail3" placeholder="(123)-456-789" name="phone">
</div>
</div>
</div>
</div>
</div>
</form>

<!- <label for="inputPassword3" class="col-sm-2 control-label">Password</label> -->
<div class="col-sm-10">
<input type="password" class="form-control" id="inputPassword3" placeholder="Password" name="pswd">
</div>
</div>
</div>
</div>
</form>

<!- <label for="inputEmail3" class="col-sm-2 control-label">Email</label> -->
<div class="col-sm-10">
<input type="password" class="form-control" id="inputPassword3" placeholder="Confirm Password" name="confirmPswd">
</div>
</div>
</div>
</div>
</form>

<div class="col-sm-offset-2 col-sm-10">
<div class="checkbox">
<label>
<input type="checkbox" name = "flgSharePermission" value="1"> Can share my email/phone to other users
</label>
</div>
</div>
3.1.3 Create Topic

It is important to know how often the user uses the application. The create topic page has six Cascading style sheets. Linking of these style sheets is very important and they are done externally to the HTML code. ‘Link’ attribute is being used to link the CSS code with the HTML code. By doing this, it becomes easy to maintain and reuse the code. The code is much organized so it is easy to edit something.

In order to create a Topic, the topic name and the options needs to be specified. For this, we first select “Add content” from the home page and then choose the topic. Several Text fields are used to accomplish this. Number of options can vary for each topic. Normally we use four options. Image of the topic can be uploaded using the “Upload image” button. The summary of the product is described in the “Body” option. When “Save” button is clicked, the HTML code will initiate the Post method in the middleware. The figure below shows the template of the create topic page.
<li><a href="#">Results</a></li>
<li><a href="#"> <?php echo $_POST['username'] ?> </a></li>
</ul>
</div> -->
<div class = "tab-panels">
<ul class = "tabs">
<li rel="panel1" class = "active">Create Topic</li>
<li rel="panel2">Ballot Items</li>
<li rel="panel3">Results</li>
<!-- username -->
<!-- <li rel="panel4"><?php echo "Not ", $_POST['username'], "." ?></li> -->
</ul>
<div id="panel1" class="panel active">
<p id= "ballotCreated">Ballot Created.</p>
<form id = "formCreateTopic" class="form-horizontal" action="afterCreateTopic.php" method = "post" >
<div class="form-group">
<label for="inputEmail3" class="col-sm-2 control-label">Email</label>
<input type="email" class="form-control" id="inputEmail3" placeholder="Ballot Name" name="email">
</div>

<div class="form-group">
<label for="inputPassword3" class="col-sm-2 control-label">Password</label>
<input type="password" class="form-control" id="inputPassword3" placeholder="Field 1" name="pswd">
</div>

<div class="form-group">
<label for="inputPassword3" class="col-sm-2 control-label">Password</label>
<input type="password" class="form-control" id="inputPassword3" placeholder="Field 2" name="pswd">
</div>
</form>
</div> -->
<p id = "ballotCreated">Ballot Created.</p>
<form id = "formCreateTopic" class="form-horizontal" action="afterCreateTopic.php" method = "post" >
<div class="form-group">
<label for="inputEmail3" class="col-sm-2 control-label">Email</label>
<input type="email" class="form-control" id="inputEmail3" placeholder="Ballot Name" name="email">
</div>

<div class="form-group">
<label for="inputPassword3" class="col-sm-2 control-label">Password</label>
<input type="password" class="form-control" id="inputPassword3" placeholder="Field 1" name="pswd">
</div>

<div class="form-group">
<label for="inputPassword3" class="col-sm-2 control-label">Password</label>
<input type="password" class="form-control" id="inputPassword3" placeholder="Field 2" name="pswd">
</div>
</form>
Figure 3.3: Create Topic template.
3.1.4 Cast Vote

Voting is the primary focus of this polling application. User votes the product based on the options available. Topic options are displayed after selecting the topic. Options present in the topic vary depending on the topic. This application allows user to edit his vote. User can change his feedback after voting.

Users can vote the product from 1 to 10, 1 being the lowest ranked option and 10 being the highest. Two or more options can have the same rank number. Once the save button is pressed, the data is stored in the database. Results are available right after the product is voted. The figure below shows the Cast Vote template.

```html
<div class="controls">
<input name="value2" class="input-xlarge focused" id="focusedInput" type="text" />
</div>
</div>
<div class="control-group">
<label class="control-label" for="focusedInput">
{
  {
    ballot.get('options').keys()[2]
  }
}
</label>
</div>
<div class="controls">
<input name="value2" class="input-xlarge focused" id="focusedInput" type="text" />
</div>
</div>
```
{  
  
  
  
  
  }  
</label>
<div class="controls">
<input name="value3" class="input-xlarge focused" id="focusedInput" type="text" >
</div>
</div>
</div>
<div class="control-group">
<label class="control-label" for="focusedInput">
{
  
  
  
  
  }  
</label>
<div class="controls">
<input name="value4" class="input-xlarge focused" id="focusedInput" type="text" >
</div>
</div>
</div>
<label class="control-label" for="focusedInput">
{
  
  
  
  
  }  
</label>
<div class="controls">
<input name="value5" class="input-xlarge focused" id="focusedInput" type="text" >
</div>
3.2 FEATURES

It is hard to give the complete picture of a tool without a video tutorial or a demo. So it is important provide screenshots. To give the readers a good sense of the application, screenshots are provided in this section.

3.2.1 Register View

To use this web application, the user has to be registered. If the user is logging in for the first time, he will not have a username or a password. He should select create an account option from the sign-in screen. Creating an account is very simple. Username and email
address has to be entered. User should pick a password to log-in to his account. The figure below shows the registration page.

![Registration Page]

**User account**

- Create new account
- Log in

- Username
- Email address
- Password
- Confirm password

**Figure 3.5: Register.**

### 3.2.2 Log-in View

Log in screen used in this application is a typical log in screen. User should enter credentials such as the username and password to log-in. This application provides the feature of remembering the username and password. Once the username and password is entered, the data is stored in the database.

Once the user is logged in, Logging out can be achieved by clicking on the image icon on the top right of the screen and then selecting log-out option. On selecting the log-out option, the application will take the user back to the log in page. The figure below shows the log-in page.
Once the user is successfully logged in, he can create a topic. Any user can create a topic as long as he is a registered to the application. To create a topic, Use should select Add content option under Navigation. Under Content, user can select different topics. After selecting a topic user should enter the name of the topic and description about the topic. User can also upload an image. Once the save button is clicked, the data will be stored in the database. The figure below shows the Topic Creation page.
3.2.4 Voting

The list of Topics can be accessed from the home page. User should be logged in to vote. User can select the topic that he wants to vote on which takes him to the voting page. Users can vote the product from 1 to 10, 1 being the lowest ranked option and 10 being the highest. Two or more options can have the same rank number.

This application allows user to edit his vote. User can change his feedback after voting.

Another important feature included in this application is adding a comment. Users can comment and can reply to others comments.

On clicking the save button, the information polled gets stored in the database. The figure below shows the Voting page.
Figure 3.8: Voting.
CHAPTER 4

TESTING

It is very important to test an application before it is launched. There are number of possible tests for this application. These are some of the testing done for this application.

4.1 FUNCTIONALITY TESTING

To make sure that no links were broken, all the links in web pages were tested, Database connectivity was tested to make sure that the data was properly stored and retrieved. Database queries were also testing.

Figure 4.1: Database connectivity testing.
4.2 Security Testing

It was important to make sure that the application was not used by unauthenticated users. Several security checks were performed like pasting internal URL directly into browser address bar without login.

![Security Testing](image)

**Figure 4.2:** Security testing.
CHAPTER 5

SUMMARY

5.1 CONCLUSION

This Online Polling web application is aimed to those users who like to get the views of other people on different topics/surveys/products. The opinions of other people will be simply the rating of different options/answers/features [7]. Many factors such as scalability, reliability, security have been kept in mind that play a crucial role in the maintenance of this web application, so that whenever there is any future enhancement, it can be done easily without any issues/trouble. Performance of the tool has been given the importance. The use of JavaScript at the client side plays a crucial role, in which the response data is returned faster than hitting the back-end database [8].

5.2 OBSTACLES FACED

Communication with the other developer, who was working on this application at the back-end and keeping everything in synchronization, happened to be a difficult task at first but we managed to get everything working together. During the times, the other developer was working on the database, I had no clue how to measure the progress of my work. Therefore, we had to arrange many meetings together so that we were sure that we were on the same page and everything was working in synchronization. Sometimes, I had to delay on certain implementations while my back end partner was working on this part.

Initially, the web application could not handle many users at one time, so sometimes there was a case that a laptop would crash and once I had to reinstall the whole server due to crash problems. Good thing is that everything worked out at the end.
CHAPTER 6

FUTURE EVOLUTION

A web application can always be improved. However it is important to make the experience of the application easy for the users to use. We can improve the UI of the application using newer and advance frameworks.

To improve the security of the application, permission can be enforced by the admin allowing the admin to choose from a list of users and request only them to vote on the ballot item. We can also assign read, write permissions to each user [9].

To make it easy for the user to access the right topic, we can categorize the topics accordingly. User should be able to choose his topic from the specific category. This can help user by reducing the burden of finding the topic in a huge list.

To improve the result display, a further enhancement would be to provide users a chart view. Users should be allowed to view the results in either chart view or the regular view. To take it one step further a 3D pie chart, or a rotating pie chart, can also be provided for the users. We can make use of google API’s to implement these features [10].
REFERENCES


