

Bachelor's thesis

Information Technology

2015

Habibul Islam

# DEVELOPING A DYNAMIC FEEDBACK APPLICATION USING JAVASCRIPT

– Responsive feedback evaluation



TURUN AMMATTIKORKEAKOULU  
TURKU UNIVERSITY OF APPLIED SCIENCES

BACHELOR'S THESIS | ABSTRACT  
TURKU UNIVERSITY OF APPLIED SCIENCES

Degree programme | Information Technology

March 2015 | 71

Instructor: Patric Granholm

Habibul Islam

## DEVELOPING A DYNAMIC FEEDBACK APPLICATION USING JAVASCRIPT: RESPONSIVE FEEDBACK EVALUATION

The purpose of this thesis project was to develop an efficient feedback application for the students who use the ViLLE e-learning environment. This thesis project aims to create feedback form contents dynamically and distribute them through a Web link. Furthermore, the student's feedback page should have a responsive design, so that it can be accessible through different mobile Web browsers offering an efficient user experience.

The application is divided into two major sections. The back end section is for the teacher, who is able to create feedback form with specific questionnaires and the front-end side is for the students, who will act as participants. Moreover, in the back end section, we have pre-defined feedback questions models, needed for the student's evaluation.

This thesis focuses on, how we can develop a personalized responsive feedback application with a well-structured questionnaire model using browser-supported programming languages: JavaScript, HTML and CSS. Although the project was specifically developed for the students feedback application, it can be used for other purposes with few modifications.

The application was tested on the local server and PhoneGap. The testing results indicate that the application can be implemented in the real system.

### KEYWORDS:

Feedback, ViLLE, JavaScript, Evaluation, Responsive, PhoneGap, Browser, Teacher's View, Student's View, Web application, User interface, HTML, CSS, JSON

## ACKNOWLEDGEMENT

I have been very fortunate to study in Finland with many wonderful people around me. During my time at Turku University of Applied Sciences, I have met lot of good teachers, whose mentorship and guidance have been instrumental in helping me develop as an IT professional.

I am very grateful to Finnish government for giving me the opportunity to study here without tuition fees. It has been an immense support for me to complete my Bachelor's degree.

Studying in Finland has always been exciting and taught me many great things for my future life. Many thanks go out to my senior fellows Dr. Nazrul Islam, Shamim Ahsan, Biplop Rahman, Md. Foyzur Rahman and Md. Ruhul Amin. Their guidance and advice has taught me many different things about life abroad, and how to approach it, and I feel myself very lucky to have friends like them.

Habibul Islam

Turku,Finland

March 2015

# TABLE OF CONTENTS

<b>LIST OF ABBREVIATIONS (OR) SYMBOLS</b>	<b>VII</b>
<b>1 INTRODUCTION</b>	<b>8</b>
1.1 The ViLLE E-learning Environment	9
1.2 Similar System / Motivation	9
1.3 Online Versus Paper Feedback System	10
1.4 Thesis Overview	12
<b>2 APPLICATION CONCEPT AND DESIGN</b>	<b>13</b>
2.1 Page Layouts	14
2.2 User Interface Design	16
2.3 Responsive User Interface	16
2.4 Teacher's Views	17
2.5 Student's Views	18
<b>3 QUESTIONNAIRE MODEL AND STRUCTURE</b>	<b>20</b>
3.1 Customization	20
3.2 Developing the Questionnaire Model and Semiotic study	21
3.3 Adding Question Type	22
3.3.1 Question Type 1: Star-rating Question	23
3.3.2 Question Type 2: Multiple Choice Question Type:	25
3.3.3 Question Type 3: Yes / No / May be	25
3.3.4 Question Type 4: Range Slider	26
3.3.5 Question Type 6: File Upload	27
<b>4 TECHNICAL REQUIREMENTS AND IMPLEMENTATION</b>	<b>28</b>
4.1 Development Tools and Objects	28
4.2 HTML 5	29
4.2.1 Developing the HTML Page for the Feedback Application	30
4.3 CSS3	32
4.4 jQuery	35
4.5 JavaScript	35
4.5.1 Implementation of JavaScript in the Feedback Application	36
4.6 JSON	40
4.6.1 Collecting Input Data Using JSON	41
4.7 Adobe PhoneGap Build	42
<b>5 TEST EXECUTION AND OUTCOMES</b>	<b>43</b>
5.1 Project Goals	43

5.2 Application Testing	43
5.2.1 Creating Demo Questionnaire (Teacher's View)	44
5.2.2 Outcomes in Large Screen ( <i>Student's View</i> )	45
5.2.3 Mobile View Test Using PhoneGap	46
5.3 Validating Input Data	47
<b>6 CONCLUSION</b>	<b>48</b>
5.4 Future Work	49
<b>REFERENCES</b>	<b>50</b>

## APPENDICES

- Appendix 1. Source codes for the HTML file (index.html)
- Appendix 2. Source codes for the CSS file (basic-style.css)
- Appendix 3. Source codes for JavaScript file (preview.js)

## PICTURES

Picture 1. Sketches of the page layout	14
Picture 2. Example of responsive user interface design	17
Picture 3. Teacher's view UI design of the ViLLE feedback system	18
Picture 4. Student's view UI design of the ViLLE feedback system	19
Picture 5. Common iconic Semiotics representation (Semiotic study)	22
Picture 6. A set of button panel in the <i>Teacher's view</i>	23
Picture 7. A part of a typical satisfaction survey rating questions	24
Picture 8. An example of a Yes/No type question (ViLLE Peer Review)	25
Picture 9. Multiple-choice question type (ViLLE Peer Review)	26
Picture 10. An example of a short answer question type	26
Picture 11. Edit and delete function for <i>select</i> type question	39
Picture 12. Demo questionnaire in the <i>Teacher's View</i> page using Firefox Web browser	44

Picture 13. Demo questionnaire in the <i>Student's View</i> page using Firefox browser	45
Picture 14. Results from mobile view using PhoneGap Build	46
Picture 15. Results from the local storage in Google Chrome browser for demo questionnaire	47

## FIGURES

Figure 1. Overall processes for the feedback system	13
Figure 2. Basic components of the main page	15
Figure 3. Five star rating question type (ViLLE Peer Review)	23
Figure 4. Multiple-choice question type (ViLLE Peer Review)	25
Figure 5. Implementation area of JavaScript in the feedback application	37

## TABLES

Table 1. Advantages and disadvantages for each of the three stakeholder groups	11
Table 2. Feedback guidelines	19
Table 3. Tools, software and coding languages used in the development process	26
Table 4. Media query's attributes and rules of use	31
Table 5. Basic elements of JSON notation and their method of use	38

## LIST OF ABBREVIATIONS (OR) SYMBOLS

<b>HTTP</b>	Hypertext Transfer Protocol
<b>HTTPS</b>	Hypertext Transfer Protocol Secure
<b>OS</b>	Operating System
<b>DOM</b>	Document Object Model
<b>CSS</b>	Cascading Style Sheet
<b>XML</b>	Extensible Markup Language
<b>HTML</b>	Hyper Text Markup Language
<b>JSON</b>	JavaScript Object Notation
<b>CDN</b>	Content Delivery Network
<b>API</b>	Application Programming Interface
<b>ECMAS</b>	European Computer Manufacturers Association Script
<b>W3C</b>	World Wide Web Consortium
<b>UI</b>	User Interface
<b>SEO</b>	Search Engine Optimization
<b>RWD</b>	Responsive Web Design
<b>CATI</b>	Computer-Assisted Telephone Interviewing

# 1 INTRODUCTION

By developing a simple application, we can accomplish many complex works in a time effective way. In today's world, we are much more dependent on internet technology. In addition to that, Web applications have pervasive influence on the interaction between machine and human due to the ubiquity of Web browsers. Browsing a Web application is no more confined within the desktop computers; it has reached its peak point through mobile devices using Web browser as a client.

In a radical way, we can say feedback is simply information. In the context of performance technology, a feedback system is often defined a means of communication by which the participant, be it a singular or a group, receives information that guides their subsequent responses, in order to obtain a significant outcome.

The main goal of this project was to create a feedback application for ViLLE e-learning environment with browser-supported programming language and relies on multi-platform Web browsers to perform the main task of this application. However, this feedback application was customized for ViLLE e-learning platform, but with this coding structure we can also develop online exam application by making some adjustments.

Although there exist several online evaluation applications, in this thesis we have tried to develop a simple responsive Web-based feedback application, which can be accessible in multi-platforms. As most of the students are using mobile devices nowadays, mobile based interface is more convenient for them to evaluate someone's project presentations, works, assignments and course feedback instantly. On the other hand, the design and the structure of the question model gives some thoughts to build modern feedback application. Moreover, the application was developed with simple but user-friendly interface, concentrating on touch screen mobile devices.

The overall application works with the collaboration of the teachers and students. It allows the teachers to create, edit and execute the evaluation form with specific contents. After the evaluation form has been created, it can be distributed to the students through a Web link and then the students access the questionnaire to perform the feedback. The application saves feedback data automatically and gather them for future analysis by ViLLE system.



## 1.1 The ViLLE E-learning Environment

ViLLE, a collaborative educational platform for University of Turku, facilitates students to participate in virtual courses and evaluate them automatically. The system was developed after carrying many research activities by the ViLLE developing team and concluded with an effective e-learning platform for students. According to the ViLLE developing team, it is considered to be an impressive learning application rather than just a Web page. It provides different courses, assignments, exercises, materials, resources, online exams, tutorials etc. to specific groups of students. The system is designed and constructed in such a way that it can be used in teaching students of all ages. It is fundamentally designed for two end users, the teacher and the students. It is possible for teachers to create virtual courses and collaborate with students providing authentic login into the ViLLE system.

ViLLE is more than just a learning environment, as it is the same tool for the planning of studies and follow-up to the teacher to the student alike. The student's learning activity and point is saved in ViLLE. These works in support of the student's own learning in mind, but also the teacher receives feedback on learning, and is able to target future teaching to action on problem areas. (The Book of ViLLE, 2014)

More details about the ViLLE system can be found at: <https://ville.cs.utu.fi/>

## 1.2 Similar System / Motivation

The common practice to administer a student survey/feedback in the most of the universities at Turku is to use Webropol survey analysis system. Webropol is an internationally recognized and Scandinavian market leader dynamic survey creation system, which is widely spread in over 30 countries around the world (Webropol, 2015). It is a third-party feedback creation application for the university, which also allows the departments to create peer review questionnaire by using their tools. Furthermore, it is a commercially distributed software system offering diverse features, for example, registrations for events, gathering feedback, reporting, text analysis, quantitative analysis, simulating and predictions, quality assessments.

However, our project was inspired by the Webropol system, where our feedback application provides only the ViLLE users a suitable tool, which can be used to develop an online evaluation. In addition to that, the proposed application aims to improve the usability

of the student evaluation more convenient and customized. It can also be determined as a similar type of feedback creation tool like Webropol, where the common goal of our application is to allow the teachers to offer the ViLLE students the opportunity to evaluate the classes and the course materials within well-suited questionnaires model.

While online feedback applications are used in numerous universities all around the world, the features of an online feedback system have to be carefully thought through and designed to make it easy for the users to use the system. There is little research literature investigating their use in education (Cummings & Ballantyne, 2000). Researchers have suggested the following elements of an effective, user-friendly, online-course rating system: ease of access and navigation; an attractive, simple, and straightforward screen; help features to assist with possible problems; confirmation of successful submission of the feedback form; and the availability of a printable feedback form in a usable format in case the form cannot be submitted electronically (Cummings, Ballantyne, and Fowler, 2001).

The development of the Web as a feedback application tool offers a number of advantages. In addition to making the survey/feedback experience smoother for the respondent, researchers also realize benefits from this technology. For example, Stanton (1998) mentioned that obtaining less missing data from a Web survey since the program can skip irrelevant questions - much in the same way a CATI system can.

### 1.3 Online Versus Paper Feedback System

Traditional method of obtaining student feedback on teaching is using paper questionnaires. However, with the advancement and availability of the latest communication technology, online feedback system mostly replaced the place of paper based feedback system. With the move to an online teaching environment, students have increased opportunities to provide feedback to staff via an online, rather than a paper questionnaire (Cummings & Ballantyne, 2000).

A research by Kelly and Marsh (1999), found that a comparison of responses between paper and online surveys show no differences. Anecdotally, lecturers in online units at Murdoch report more considered comments from students in electronic discussion groups. The experience at Murdoch shows no discernible difference in ratings of teaching between

paper based and online surveys. Confidence in the technology is an important concern for survey administrators.

One of the major advantages often claimed for electronic surveys is the minimal cost. A number of researchers have suggested that e-mail surveys cost less than mail surveys (Bachmann & Elfrink, 1996; Kiesler & Sproull, 1986; Parker, 1992; Schaefer, 1998; Sproull, 1986). It can be said that electronic surveys reduce paper waste.

Once a survey is online, there is a rise in expectations that it will work correctly and at a faster rate. At the user interface respondents will quickly become frustrated if they are unable to complete their questionnaire quickly and submit it successfully. According to the observation of Rick Cummings & Christina Ballantyne at Murdoch University mentioned that there are three major stakeholder groups in the student evaluation of teaching feedback process – the teaching staff, the students and the survey administrators. The level of computer literacy among students and among staff is also an important issue when considering transferring surveys online. Table 1 summarizes some advantages and disadvantages for each of the three stakeholder groups.

Table 1. Advantages and disadvantages for each of the three stakeholder groups. (Cummings & Ballantyne, 2000)

Stakeholder Group	Advantages	Disadvantages
<b>Teaching staff</b>	<ul style="list-style-type: none"> <li>• Save class time</li> <li>• Increased flexibility in content and administration</li> <li>• Quicker turnaround time</li> </ul>	<ul style="list-style-type: none"> <li>• Potential for low response rate</li> </ul>
<b>Students</b>	<ul style="list-style-type: none"> <li>• Increased flexibility in when and where to respond</li> <li>• Makes use of popular current technology</li> <li>• Increased anonymity in written responses</li> <li>• Can include students not attending class on a particular day</li> <li>• Quicker turnaround time</li> </ul>	<ul style="list-style-type: none"> <li>• Potential for low response rate</li> <li>• Poor information technology literacy and access among some groups of students</li> <li>• Increased reliance on technology</li> </ul>
<b>Survey administrators</b>	<ul style="list-style-type: none"> <li>• Less handling of forms</li> <li>• Fewer errors in data entry</li> <li>• Reduced cost of materials</li> <li>• Reduced data entry costs and time</li> <li>• Greater flexibility</li> <li>• Potential to fully automate the survey process</li> </ul>	<ul style="list-style-type: none"> <li>• Potential for low response rate</li> <li>• Uncertain cost for initial set-up</li> <li>• Increased reliance on technology</li> </ul>

## 1.4 Thesis Overview

The structure of this thesis is divided into six chapters.

Chapter 1, Introduction, gives a clear overview of the thesis. This chapter explains the goal for the project as well as the motivation for the project.

Chapter 2, Application Concept and Design, introduces key features of this application. It explains key concept of the page layout and describes how it was implemented in the project.

Chapter 3, Questionnaire Model and Structure, provides an introduction of the questionnaire types used for the application. It also provides some theoretical backgrounds related to application development procedures.

Chapter 4, Technical Requirements and Implementation, mainly provides the coding solution for this project. It shows the main structure of the code along with theoretical knowledge.

Chapter 5, Test Execution and Outcomes, deals with the test design, implementation, and running the test cases in the Web browsers. It also shows the test validation of input data representation in local server. The test case result of small devices is presented using PhoneGap.

In chapter 6, Conclusion, is the synopsis of the thesis. It describes the main prospects of the thesis project.

## 2 APPLICATION CONCEPT AND DESIGN

A well-structured concept or a well-built design sometimes can be a pioneer in terms of *application architecture*. Application architecture refers to the behavior of an application. It concentrates on the interaction with the user. Before developing the concept and design of the application, it is substantial to know the purpose, aims and goals, potential users and developing methods.

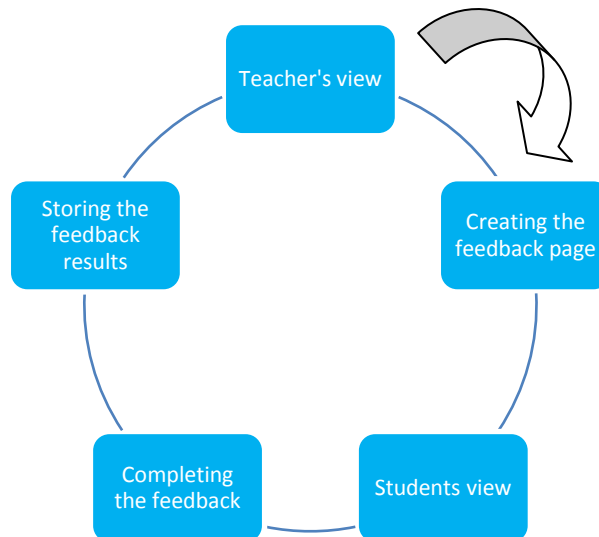


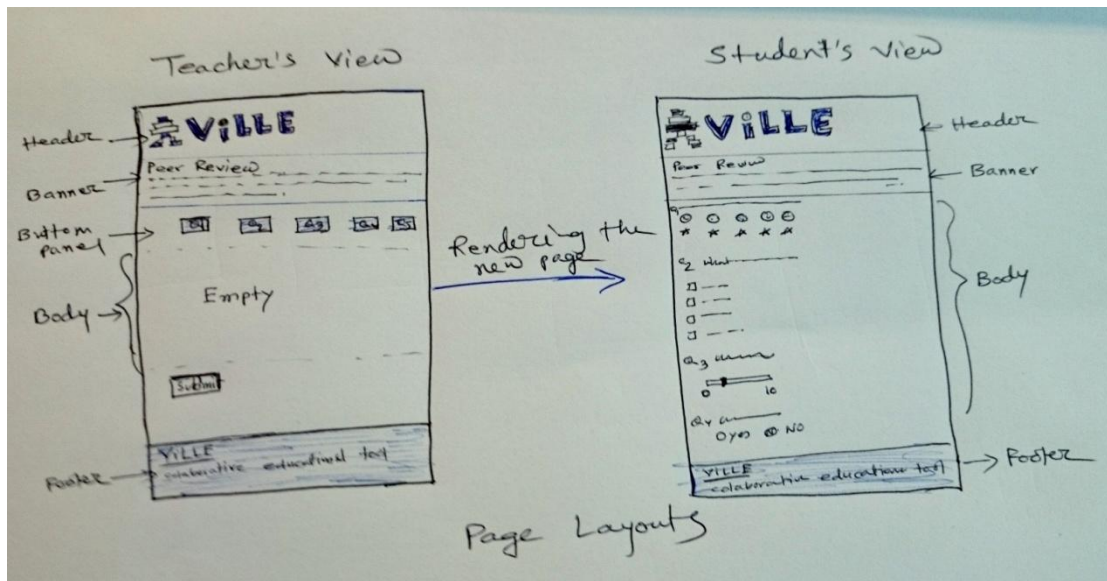
Figure 1. Overall processes for the feedback system

The core work for this project is to design a user friendly and well-structured feedback application, which can be accessible through different devices maintaining responsive user experience. The concept developed for this project is the confluence of two end users view: *Teacher's view* and *Student's view*. After logging into the teacher's domain of ViLLE platform, the teacher is able to create the feedback form using predefined question types. In *teacher's view*, the teacher is able add, edit and delete different question types and provide the auto generated *URL to the* students.

The students are the participants group for the feedback. They can access it by logging to the ViLLE student's domain and complete the feedback by pressing the submit button at the end of the page. In here, we have an authentication login system for ViLLE users, but in general, it can be distributed through the link without authentication if it is in a publicly accessible domain.

## 2.1 Page Layouts

The page layout serves the arrangement of visual elements of the Web pages. It is one of the most important sections while designing the application's user interface. It can be used as a guideline of the whole work. While approaching the topic of designing the page layout, new designers or developers make the common mistake of starting directly from Photoshop. After developing the application concept, it is always effective to draw the page layout in pen and paper. It not only helps to ameliorate the work but also works as a map.



Picture 1. Sketches of the page layout

In our approach, we have designed a very simple layout with all the basic elements we need to complete the task. As we have two different views for this application, we have to consider two layouts. By using JavaScript, we can actually render the *Student's view* apart from the *Teacher's view* page. Here, in total, we need five basic page layout components for two different views, which are mentioned below:

- Header
- Banner
- Navigation button Panel
- Body
- Footer

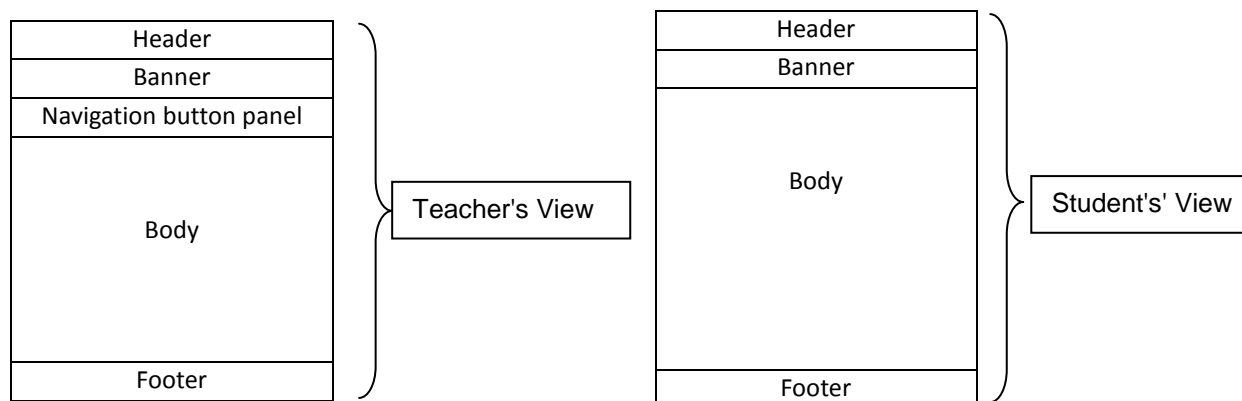


Figure 2. Basic components of the main page

Firstly, the header is usually placed at very top of any Web page. Its basic task is to clasp the logo. The header section, as like other Web pages, it displays the logo of the ViLLE platform with white transparent background.

The Web page banner is generally reserved for advertisements or set of instructions. In the banner section, we will have default feedback instruction texts for the students. It covers the basic information about point of scale for each question type. Before performing the feedback/evaluation, the participant will receive some brief information about his/her task. While the teacher is creating the feedback forms, he/she is able to customize the banner section according to the feedback requirement.

Navigation buttons are a set of buttons or small graphical images usually placed in a row or column in the Web page to navigate different components. They are also used to render specific elements in the same page. We have a set of question type's button panel only in *Teacher's view* page. The teacher can add questions by clicking the specific question type from the panel.

The Web page body section is the place where the bulk of question types shall be added. In *Teacher's view*, the body section initially blank before adding any questionnaire by pressing the button panel. Finally, the footer is the bottom section of any Web page. Generally, it contains copyright notice, contact information and link to privacy policy.

## 2.2 User Interface Design

The term *User Interface*, refers to a space where interaction between humans and machine occur. It is the visible part of any applications or programs.

It concentrates on the user's experience of the following areas:

- Websites
- Software applications
- Mobile communication devices
- Computer appliances
- Machines

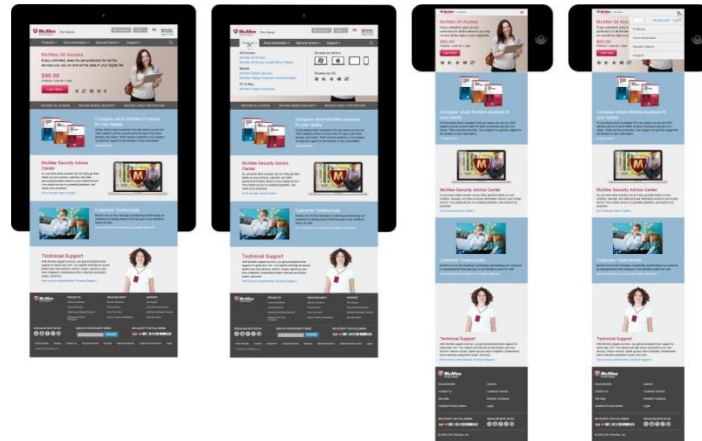
In addition, an effective operation and control of the machine on the user's end and response from the machine, is the main objective of a good user interface design. A well-designed user interface facilitates the user to complete the task without paying extra attention. As we have different Web user platforms nowadays, it becomes more challenging for Web designer to design a user interface for multi-channel users. Moreover, we have a growing number of mobile users, and in order to perform the interaction with those users, designing a responsive Web interface is very advantageous.

## 2.3 Responsive User Interface

*Responsive user interface* means to display a Web application in different style properties, based on the screen size, orientation, resolution, color capability and other characteristics of the user's device. In Web designing approach, responsive Web design (RWD) intends to craft a site by providing optimal viewing experience to its user (Responsive Web Design, Wikipedia, 2014). In responsive UI, a Web page is able to accommodate itself on the device it is being displayed on. Nowadays, responsive user interface design is very useful in order to serve a wide range of users. Some of the benefits of RWD are given below:

- Effortless multi-device adaptation
- Easy to maintain
- Improves SEO
- Seamless user experience



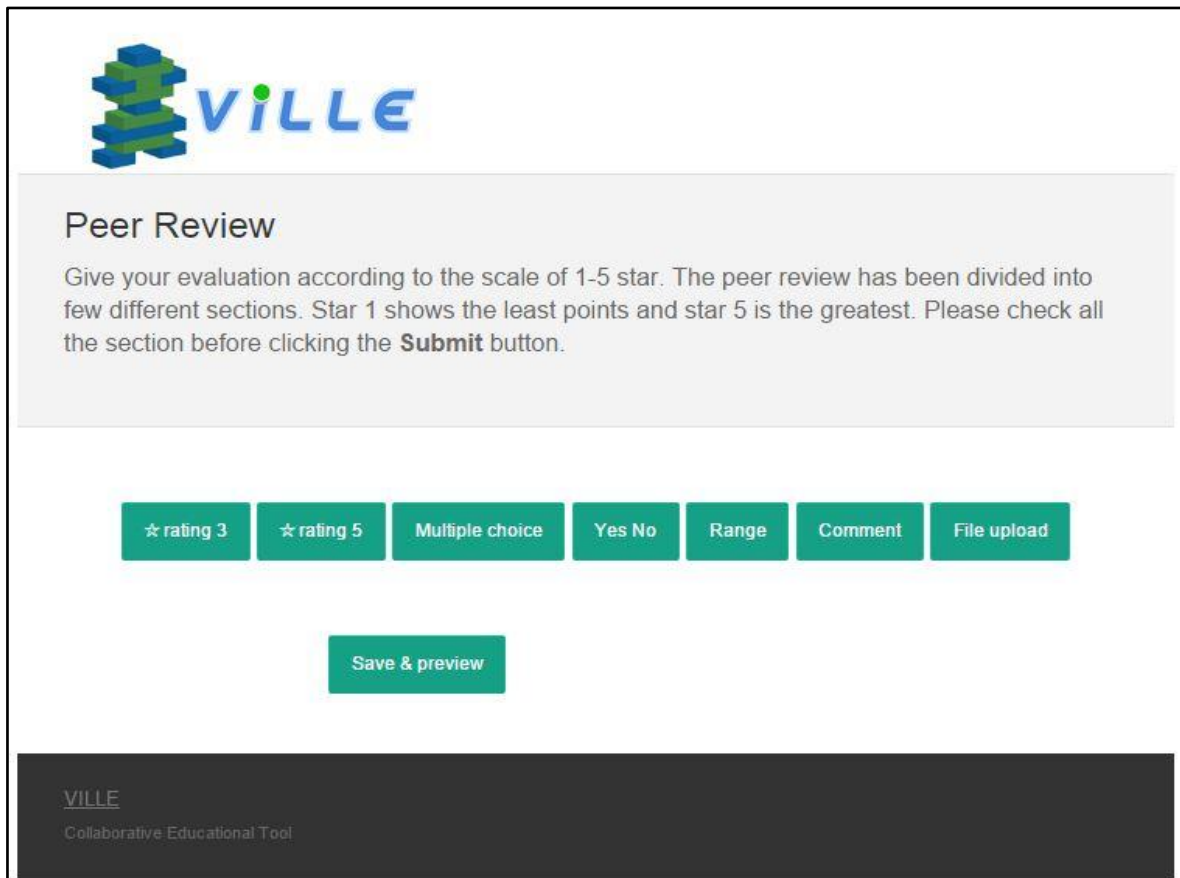


Picture 2. Example of responsive user interface design

## 2.4 Teacher's Views

The viewpoint of an application is usually application-specific and depends extensively based on the application domain. We have divided our system into two different user interfaces. In *Teacher's view* that addresses a variety of issues, including: creating questionnaire, editing, deleting and generating the preview. The term '*Teacher's View*' refers to the user interface of the teacher in VILLE feedback application.

As in the page layouts mentioned earlier, we have created a simple user interface for the teacher. In *Teacher's view*, the page comes with header, banner, button panel, body (initially it is blank) and footer components along with the preview button in the center of the body section. Moreover, this page stands as our basic layout for both user interfaces.



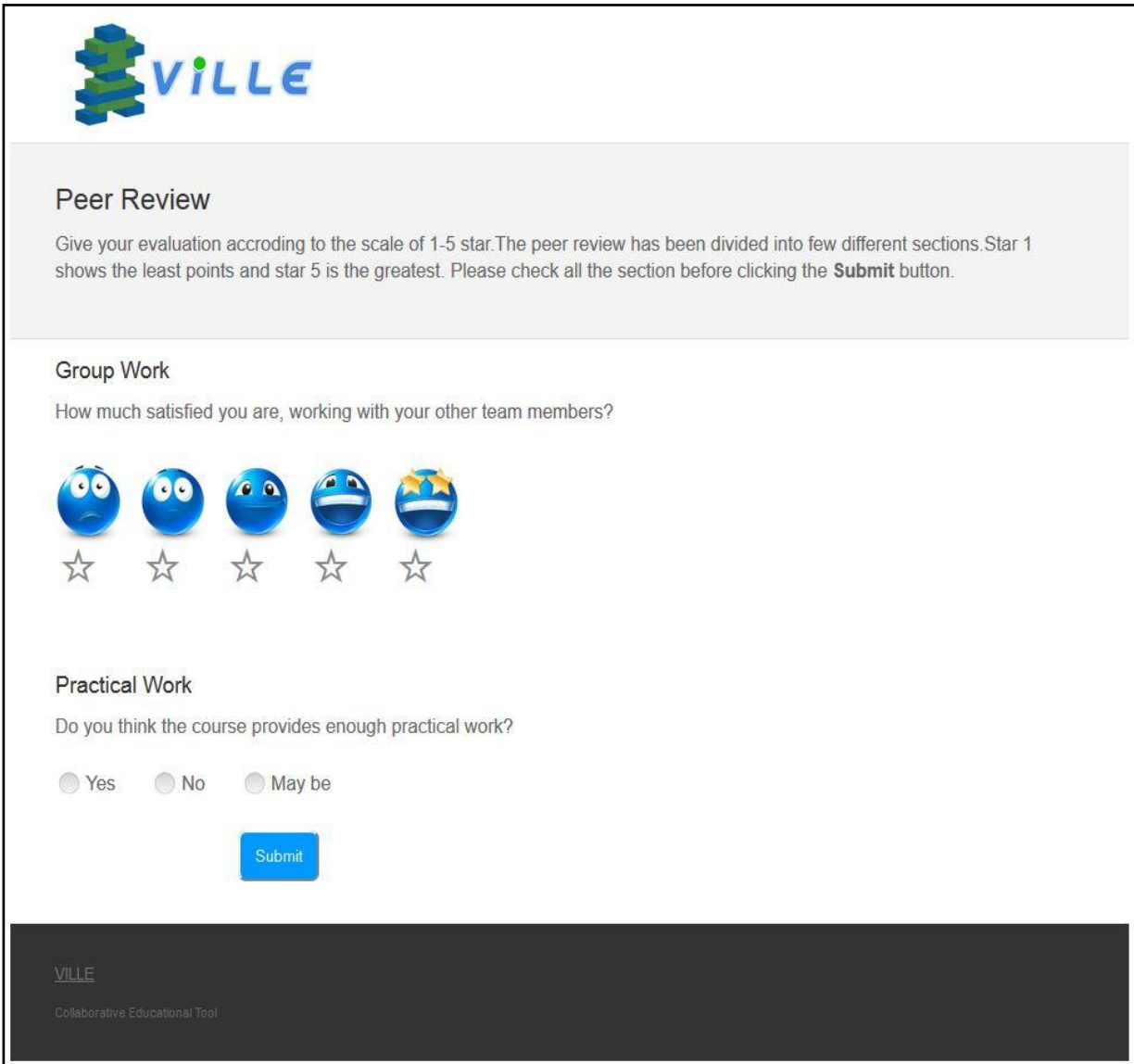
Picture 3. Teacher's view UI design of the ViLLE feedback system

The action performed in the teacher's view UI is to create the questionnaire dynamically by adding them into the body section from a button panel. Here we have a set of question type in the panel. Later the next chapter [Chapter 3], will describe question types in detail.

## 2.5 Student's Views

The *Student's View* is actually a part from Teacher's View. It is dynamically created from *Teacher's View*. Here is the crafty part of this application, which is rendering by using JavaScript. JavaScript allows rendering a new page from an existed page by defining specific page elements. The term '*Student's View*' refers to the user interface of the students in ViLLE feedback application.

*Picture 4* represents the state of *Student's View* after creating the questionnaire by the teacher from *Teacher's view*. As the page layout for *Student's View* defined earlier in *section 2.1*, it displays the page along with header, banner, body and footer components.








**ViLLE**






### Peer Review

Give your evaluation according to the scale of 1-5 star. The peer review has been divided into few different sections. Star 1 shows the least points and star 5 is the greatest. Please check all the section before clicking the **Submit** button.

#### Group Work

How much satisfied you are, working with your other team members?

#### Practical Work

Do you think the course provides enough practical work?

Yes
  No
  May be

ViLLE  
Collaborative Educational Tool

Picture 4. Student's view UI design of the ViLLE feedback system

However, the page may contain many questions, and in order to view them, the students will have the scrolling up and down option. The use of relevant images in the Web-based feedback form is an important issue. Proper use of images, can contribute to build up a well-designed questionnaire model. For example, Bishop's (1997) Web survey involving landscape picture perception used color images, demonstrating one of the advantages of the graphically enabled Web survey.

### 3 QUESTIONNAIRE MODEL AND STRUCTURE

One of the most common and effective approaches of application development is to create a prototype. A *prototype* is the sample model of a product that is employed as a trial to test the concept through the end of the actual development process. In few workflow models, contriving a prototype (sometimes called **materialization**) is the action between the formal specification and the evaluation of an idea (Prototype, Wikipedia 2014). Conducting a feasibility study is also very important before establishing the final model and structure of any application.

Before going through our actual development process, we should answer the following questions:

- Is this a unique idea?
- Is there any similar kind of application exist?
- What are the strengths and weaknesses of the existing application?
- What is the traditional approach to create this kind of application?
- Who are the main users?
- Which platform we are developing for?

These questions help application developers to describe and manipulate their own project at a high level as relations, while giving control of how relations are represented substantially in the project.

#### 3.1 Customization

A custom application, an application that already exists, is especially developed for some specific organization or other user. To address this, we have developed a customized feedback application for ViLLE users. In order to deal with this approach, we have gone through some of the comparable feedback applications available in online.

Most of the feedback applications, for example: customer feedback, product review or a survey, follow a common format which is a universal approach to handle feedback process. Furthermore, a variety of such feedback applications are available in online, each with the own recognized its strengths and weaknesses. For such a multifunctional purpose, one specific questionnaire methodology is not suitable for all kinds of projects. Each of the available feedback application may be best suited for particular project.

However, a customized application is usually built to enhance richer interaction with a specific group of users. It can be built following a similar kind of system by modifying with different layout and style.

### 3.2 Developing the Questionnaire Model and Semiotic study

In the feedback system, a questionnaire is a series of questions, gathering information from respondents to deal future outcomes.

A complicated or confusing questionnaire model often fails to yield enough responses from the respondents. To overcome this failure, the work involves developing a questionnaire structure, is very challenging task for a developer. According to Tosti (1986), there are three general principles for effective feedback : fit, focus and timing. The table below provides guidelines for creating an effective feedback system:

Table 2. Feedback guidelines (Feedback Systems by Tosti, 1986)

Fit	Focus	Timing
<ul style="list-style-type: none"> <li>• Ensure the respondent can understand and act upon the feedback</li> <li>• Tailor the information to the performer's needs. Match praise or rewards to what is important to the individual</li> <li>• Use the performer's language(especially with novices)</li> </ul>	<ul style="list-style-type: none"> <li>• Concentrate on the behavior, not the person.</li> <li>• Avoid overloading the performer. Focus on the area or subject at a time</li> <li>• Be specific</li> <li>• Check receptivity</li> </ul>	<ul style="list-style-type: none"> <li>• Give feedback when the performer can use it. Encourage now (summative) and advise later (formative).</li> <li>• Give feedback often enough. ( more often is particularly important if low confidence is an issue or the behavior has the potential for severe consequences)</li> </ul>

On the other hand, human interaction with computer and mobile devices have been developed remarkably over the past few years. Nowadays, text is replaced by a set of symbolic icons or signs in computer and mobile user interfaces. It can be effective to enhance the interaction with humans by using symbol and signs rather than small and long text in computer user interfaces. Human performance over Semantics (the study of meaning like phrases, signs, icons and symbols) is another issue to be addressed while developing an application for mobile users. In addition to that, in our feedback system, we tried to implement knowledge of Semantics, by assigning icons in some of the question types to make the interaction more convenient for mobile users.

Furthermore, the study of meaning making of signs and symbols is referred to as Semiotics (also called semiotic studies). A symbol or an icon stands for a meaning according to its method of use and right placement. As a result of this, the graphical user interface in computer and mobile devices is much more occupied with symbols than plain text. Semiotics is largely involved in Web applications as well.



Picture 5. Common iconic Semiotics representation (Semiotic study)

To find out the right icons in order to represent specific information or activities is a challenging and tough assignment for a Web application developer. Recognition of icons and also interaction with them depends on the experience of the user by using Web interface for a long time. However, in our feedback application, we focused on making the question contents visually clear and assigned an icon where it is convenient to provide a better user interface.

### 3.3 Adding Question Type

From the buttons panel in the *Teacher's View*, the teacher controls the construction of his/her own questionnaire from the pre-created question type. In some other online

feedback system, the question-adding action performed from a drop-down menu, where the user needs to put extra attention.



Picture 6. A set of button panel in the *Teacher's view*

Here, in our system, we placed the question-adding buttons panel at the top, so that the user can easily perform the action at a first sight. Every button starts with a '+' sign, which indicates to the user that the question-adding action can be performed from this button. All buttons are interactive and as a result, when the user clicks on a particular button, it adds that question type in the body of the page. The user (Teacher) can add as many questions as he/she needs to build up own questionnaire. The questions are added one by one, without overlapping each other.

### 3.3.1 Question Type 1: Star-rating Question

Stars refer to symbols in the Web interface context. They are often used for classification purposes. For example, one to five stars are used by the reviewer to express their satisfaction level according to the scale given for a particular section. Stars provide some benefits in rating systems. Firstly, a star is most common and well-recognized symbol for the Web users. Secondly, it is not only user friendly but also very effective to analyze satisfaction level.



Figure 3. Five star rating question type (ViLLE Peer Review)



Varieties of such customer/student satisfaction rating questions have developed over the years, each with its own characteristics. As we followed the Semiotic methodology in our application development process, Emoji icons (facial expressions in graphical image) along with star ratings can be an effective combination for the respondent. Figure 3 above, shows the structure of the five star rating question. When the reviewer selects any star, it will automatically fill with color. Star rating one shows the least satisfaction level, whereas star-rating five is considered excellent. We have developed this star rating question by using CSS style. Those Emoji icons represent the satisfaction level of the user instead of using long text.

**Please rate your level of agreement with the following statements**  
One answer for each line.

	Strongly agree	Agree	Disagree	Strongly disagree	No opinion
We understood your enquiry or problem	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
We gave you the feeling that we cared about you	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
We gave you the confidence that we could resolve the matter	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Our staff were clear and easy to understand	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Our staff were patient, helpful and efficient	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
We followed up our promises	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
We met the timescales we indicated to you	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>

Picture 7. A part of a typical satisfaction survey rating questions

In typical survey rating methodology, we may see that a set of questions are arranged together in a row and column. The approach advocated in this typical survey-rating question is based on desktop computer user. While applying this type of (Picture 7) questionnaire model for the multiple devices (responsive view), it is usually complicated for the mobile user to interact with those questions because of complex view. In order to deal with this problem, we have arranged our question structure into few different sections, so that the reviewer can easily go through each of the questions.



### 3.3.2 Question Type 2: Multiple Choice Question Type:

Multiple choices are selected from a pre-defined list where respondents are asked to select preferred answer out of the choices. The multiple-choice format is mostly used in the education sector. In the feedback questionnaire model, multiple-choice is often very effective and required question type in order to evaluate a category. In our question model, it allows the teacher to add list of choice and modify them according to the demand for the specific task. When the respondent checks the box, it is marked with '☑' sign.

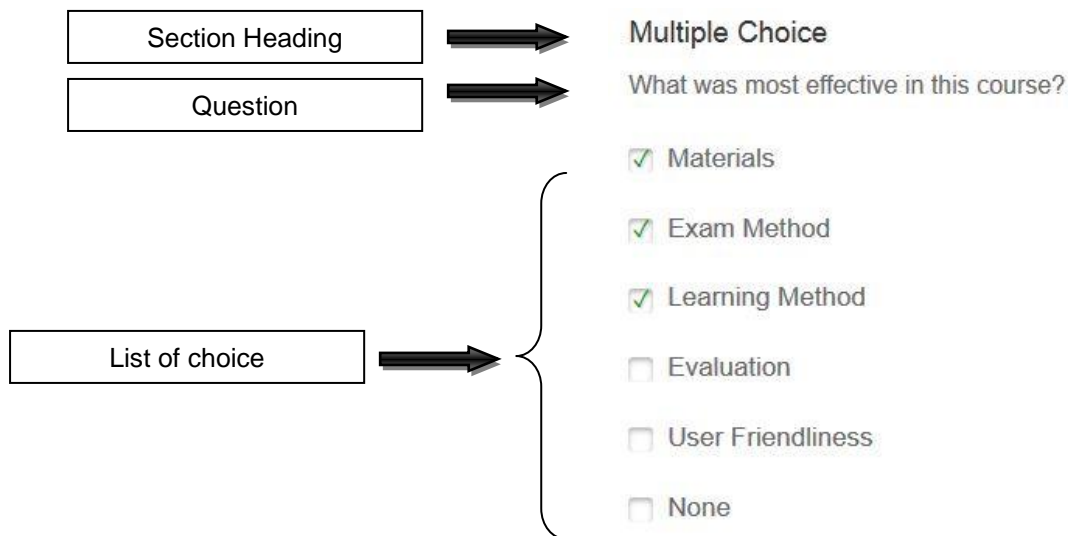


Figure 4. Multiple-choice question type (ViLLE Peer Review)

In addition, the *None* option also provides an extra control over the selection. If the *None* option is selected, it unchecks all the choices made earlier by the respondent. JavaScript allows extra control over other elements to perform this action. Moreover, in the touch screen mobile devices the options selection can be performed by finger tapping.

### 3.3.3 Question Type 3: Yes / No / May be

**Future Study**

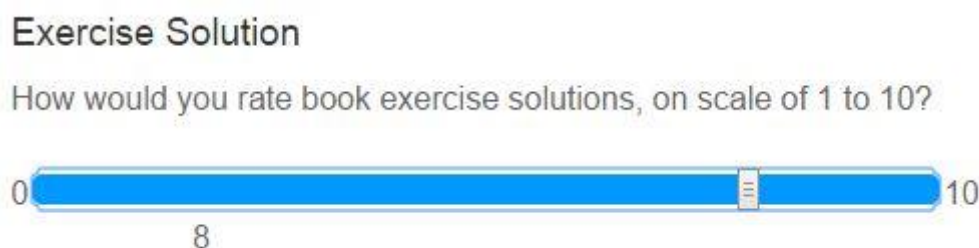
Would you like to take some other courses from this teacher in the future?

Yes     No     May be

Picture 8. An example of a Yes/No type question (ViLLE Peer Review)

Yes/No or May be, questions are usually labeled as closed question because they have only one option to select out of three options. In linguistics, it is referred to as a polar question. In the feedback system, this polar question type adds benefit to find out one straight answer from the respondent. In our project, we have layered this Yes/No answer button with CSS3 style to enhance the user experience.

#### 3.3.4 Question Type 4: Range Slider



Picture 9. Multiple-choice question type (ViLLE Peer Review)

Among many other question types, the Range Slider can be very effective and user friendly tool for the evaluation system. It lets the user to input a changeable range between pre-defined scale by adjusting it frequently. The main advantage of a range slider is an interactive performance. It can be used in numeric and non-numeric scale. It is also possible to get the input value in a decimal format. For example, respondent can also point out 7.83 on 1-10 range. Moreover, a range slider can be used in various ways by defining its position and behavior by the developer.

#### Question Type 5: Short Answer Question

##### Comments

Please write your comments if you have something else to add?

Picture 10. An example of a short answer question type (ViLLE Peer Review).

Short answer questions are set to receive anything beyond the frame of the pre-defined options in the feedback questions. In addition to that, responses are not limited in length, but they let students know of the teacher's expectations.

### 3.3.5 Question Type 6: File Upload

The file upload option was added as a supplementary option for the students. Just in case teachers need an attachment of any solution of an exercise. To deal with this additional requirement, the file upload option has been added in the panel.

## 4 TECHNICAL REQUIREMENTS AND IMPLEMENTATION

The technical requirements refer to the technical perspectives that the system must fulfill in order to execute the application properly. As our application is based on a browser supported programming language, the applicable programming source codes and images are prior to technical solutions.

To run the application successfully, proper arrangement of the source files is a prerequisite for the implementation process. Each external source code files must be placed in a proper manner, which is defined by the base file: *index.html*. The *index.html* file links external CSS, JavaScript and image source with predefined path while executing the program. Generally, external files are defined into the *head* section of the base html file.

The basic technologies that were used to develop this feedback application are: *HTML5*, *JavaScript* and *CSS3*. PhoneGap shows the test environment for the mobile users. Additionally, with the identified functional requirements in the preceding section, several images are also essential to implement the program. In *JavaScript* code, using special JavaScript libraries like *jQuery* and *jQuery Mobile* provides extra functionalities for the application.

### 4.1 Development Tools and Objects

According to the application development process, we proposed similar kind of approaches like any other Web-based application requires. However, for developers, many different tools and frameworks can be used in order to aid in the development of an application. A brief introduction to all the technologies mentioned is provided in the following subsections.

Table 3. Tools, software and coding languages used in the development process

Name	Type	Description
JavaScript	Language	An object-oriented computer programming language
jQuery	Library	A lightweight JavaScript library.
HTML5	Language	A markup language of the Internet used for structuring and presenting content for the World Wide Web
CSS3	Language	A style sheet language used for describing the look and formatting of a document written in a markup language.
Adobe Photoshop CS6	Software	Image-editing software from Adobe.
WebMatrix 3	Software	A text editor application tool
PhoneGap	Framework	A mobile development framework
Desktop, Laptop, Smart Phones and Tablets	Device	An electronic device which is capable of receiving information
Firefox, Chrome, IE, Safari	Browser	A computer program that used to navigate the World Wide Web

To build a Web based multi-platform application, it is requisite to use Web standards languages like HTML, JavaScript and CSS. Besides, coding other corroborative tools like software or application program, pave the way to the harmony in the application development procedure.

Image editing software like Adobe Photoshop CS6 is also required for the project. To write codes we have used Microsoft WebMatrix-3 text editor. Along with popular browsers testing, the mobile user compatibility was also implemented through PhoneGap, an application that also leads to create mobile apps using standardized Web APIs.

## 4.2 HTML 5

HTML stands for Hyper Text Markup Language, which is an imperative markup language for characterizing Web documents. In addition, HTML5 is the fifth and final revision [update October 2014] of the HTML standard of the World Wide Web Consortium (W3C), the organisation responsible in designing and maintaining the language (W3schools [2], 2014).

According to W3C, Hypertext is text which contains links to other texts. Hypertext is the approach by which we surf around on the Web — by clicking on hyperlinks which proceed to the next page.

HTML consists a series of short codes, which work behind the scenes of modern Web browser like Firefox, Chrome, Safari and so on. The HTML codes are formulated with a set of markup tags, then saved as a html file and can be accessed the viewpoint through Web browsers, which reads the file and render the text into a visible form. By writing own HTML statement using correct tags, leads the developers to build their own version.

According to W3Schools explanation, here are some most obvious tags used in HTML documents:

- The DOCTYPE is the Document Type Definition (DTD), declares a validator which version of HTML to use in checking the document's syntax
- The text between <html> and </html> represents the Web document
- The text between <body> and </body> represents the visible page content
- The text between <h1> and </h1> represents a heading
- The text between <p> and </p> represents paragraph

HTML markup tags are the main elements of html documents. HTML tags mostly placed in a pair like <title> and </title>, and together in a sequence represents the whole HTML page.

For more details about HTML5, the best resource can be found at: <http://www.w3schools.com/html/default.asp>

#### 4.2.1 Developing the HTML Page for the Feedback Application

HTML files are defined with the *.html* extension. In our project, *index.html*, describes how to represent relations as a combination of JavaScript and CSS in order to accumulate page layout structures (markup). Our target is to prove that how successfully we can render the 'Student's View' page in representation of feedback questionnaire by making the relation with 'Teacher's View' in *index.html* file. The file consisted with a set of HTML tags follows the HTML5 rules and combination.

## Viewport settings

The use of Meta tag is not actually W3C stipulation, as a result of comprehensiveness of iOS devices and WebKit, it has been in a dominant position by public acceptance. However, in short, they offer a number of ways to control how an entire Web property displays and behaves on a range of mobile devices (Frain, 2013). To define the display of the Web property at the device width with the initial scale of 1, in context of responsive view, we can initialize the following instructions in <meta> tag:

```
<meta name="viewport" content="width=device-width; initial-scale=1.0">
```

## Linking the external files

The HTML <link > tag defines the link between .html files and other external resources in order to establish the relationship with each other. For example: to link the external CSS file called *basic-style.css* under *css* folder we can refer it in <link> tag.

```
<link rel="stylesheet" href="css/basic-style.css">
```

Similarly, to define the client-side script such as JavaScript, which contains scripting statements, we can refer them in following way:

```
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1/jquery.min.js"></script>
<script src="js/libs/modernizr-2.6.2.min.js"></script>
<script src="http://code.jquery.com/jquery-1.7.1.min.js"></script>
<script src="js/preview.js"></script>
```

In order to use the jQuery CDN, the reference of *http://code.jquery.com* in the scripting tag links with the content delivery network.

## Use of <div> tag in HTML

The purpose of the <div> tag is to provide the guidance of a section or division in an HTML document. The <div> is tag mostly used along with CSS to organize block-elements in order to style them with CSS properties. It is acknowledged as a typical method for classifying the axiomatic sections of a document.

The HTML below shows two 'divs' being used in our HTML document in association with id attributes (which already defined by CSS properties in external CSS file) to identify different sections:

```
<div id="banner">
  <div id="logo"><a href="ville.html"></a></div>
</div>
```

## Creating the Teacher's View using HTML5 elements

We have defined HTML tags in different sections of the page content in corporation with ID attributes of CSS. According to the *Teacher's View* user interface, the following markup shows fundamental tags used in our *index.html* document

```
<!DOCTYPE html>
<html>
<head>
  <title>.... </title>
<!-- Mobile viewport settings -->
<!-- External CSS link-->
<!-- jQuery library link -->
<!-- JavaScript source link-->
</head>
<body>
  <div> </div> <!-- #Banner area -->
  <section></section> <!-- #Button Panel -->
</body></html>
```

The full source code can be found in **Appendix 1.0**.

### 4.3 CSS3

CSS3 is the latest standard of Cascading Style Sheets (CSS), which represents the look and style of documents written in HTML. HTML has its limitations when it comes to layout, whereas CSS provides more control over the Web pages by implementing exact measurement, layout and design. For example, HTML allows to create six different level of heading and fonts, and also has table which can be accommodate over alignment. This could be a best solution while creating an ordinary looking Web document. However, with CSS, we can get much better control of our Web document and we can specify exactly how big a font size will be, exactly where we can place an element on a page. Moreover, the CSS provides different functionality over Web pages and it is divided into particular "modules". According to W3Schools (W3schools 2014) some of the most important modules include:



- Selectors
- Box Model
- Backgrounds and Borders
- Image Values and Replaced Content
- Text Effects
- 2D/3D Transformations
- Animations
- Multiple Column Layout
- User Interface

Each of the modules has their own properties. As an example, in backgrounds and borders module, *the background-image, background-position, background-color, background-size* properties set the element's background image, position, color and size respectively.

### **CSS media queries for responsiveness**

Specific CSS design intends to characterize the Web content for tiny screens, giant screens and anywhere in between, whereas media queries allow us to make a responsive perspicacity. The media query syntax permits for the creation of rules, which will be applied counting on device characteristics. The media query below defines the minimum width of the content for large viewing, possibly full screen viewers.

```
@media only screen and (min-width: 1240px) {  
}
```

In CSS responsive properties, there are certain different items we can query. The most often-used media query's attributes and their purpose of use are given in the table below:

Table 4. Media query's attributes and rules of use (Web Fundamentals by Google Developers, 2014)

Attributes	Rules
<code>min-width</code>	Rules applied for any browser width over the value defined in the query.
<code>max-width</code>	Rules applied for any browser width below the value defined in the query.
<code>min-height</code>	Rules applied for any browser height over the value defined in the query.
<code>max-height</code>	Rules applied for any browser height below the value defined in the query.
<code>orientation=portrait</code>	Rules applied for any browser where the height is greater than or equal to the width
<code>orientation=landscape</code>	Rules for any browser where the width is greater than the height.

Besides, there are other specific media queries for different devices and that should be associated with CSS document to provide the actual responsive experience for multiple devices. All the essential media queries have been provided for our project, which also mentioned in *basic-style.css* document in **Appendix 2.0**.

## Responsive images

As the width attribute is not supported in HTML5, so instead of referring an absolute width value by a HTML attribute in the tag of an image, assigning the CSS rule that targets the image as a percentage relative width value like the example below, makes the image responsive.

```
img {
  max-width: 100%;
  height: auto;
  width:100%;
}
```

What this will do is make the image display 100% of its size within its parent element available width space. For Safari browser it needs to define the width to auto, whereas the rest of the browsers are justified with width:100%.

#### 4.4 jQuery

jQuery, a fast and concise cross-platform JavaScript library, allows Web developers to perform extra functionality in the scripting situation. It performs many different tasks like, access elements in a document, alters the content of a document, modifies the appearance of a Web page, responds to a user interaction and simplifies retrieving information from server without refreshing a page. Moreover, the main feature of jQuery is that it simplifies common JavaScript task.

jQuery supplies powerful methods for changing and manipulating HTML elements and attributes. One very important part of jQuery is the possibility to manipulate the DOM (Document Object Model). According to W3C the DOM defines a standard for accessing HTML and XML documents:

"The W3C Document Object Model (DOM) is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document." (jQuery-Get Content and Attributes, w3schools, 2014)

jQuery provides three useful methods for DOM manipulation are:

- `html()` - method sets or returns the content of designated section (Mainly HTML markup)
- `text()` - method defines or returns the text content of designated section
- `val()` - method is used to set or returns the value of form fields

#### 4.5 JavaScript

JavaScript, a dynamic programming language introduced by Netscape, allows Web developers to build interactive Web document. It is most typically used as a part of Web browsers to enable client-side scripts to act with the user and reform the document content that is displayed. According to Wikipedia, JavaScript uses a number of identical ideas from Java, an object-oriented programming adopted from C++, where the syntax (set of rules) is more similar to a high level programming language C and is based on ECMA Script (JavaScript, Wikipedia, 2014).

As a client-side scripting language, JavaScript mostly represents itself by processing the source code with client's server instead of synchronizing the Web server, meaning that Web page can be loaded by JavaScript functions without communicating the Web server. For example, it can react with user for instance showing message instantly, without retrieving the data from the Web server.

A variety of implementation areas for instance Web development, game development, mobile and desktop application development has involved JavaScript in a recognized area. In addition, JavaScript is frequently used to create Web based applications, which does a great advantage for the developer in cross-platform development by compelling less written code.

JavaScript offers immense benefits over the development of information technology. According to Web standard curriculum of W3C, some of the important features of JavaScript depicts below (What can you do with JavaScript, W3C, 2014):

- JavaScript is extremely simple to implement. We just need to place code within the markup language document and tell the browser that it is JavaScript.
- It works on internet user's computers — even after they are offline.
- It permits to make extremely responsive interfaces that improve the user response and supply dynamic performance, while not having wait for the server to react and show another page.
- It will load content into the document if and once the user desires it, while not reloading the complete page — this is often usually mentioned as Ajax.
- It can check the potentiality of a browser and react consequently — this can be known as Principles of unnoticeable JavaScript or generally defensive Scripting.
- It can facilitate fixing browser issues or patch holes in browser support — as an example fixing CSS layout problems in certain browsers.

#### 4.5.1 Implementation of JavaScript in the Feedback Application

The entire work of this feedback application predominantly depends on JavaScript's mechanism. However, the work described above covers the designing phase, where we have a set of buttons to be performed in order to manipulate particular task. To address

this specific function, we have integrated JavaScript into our system to perform the operation. Using JavaScript we can implement the following task in our application:

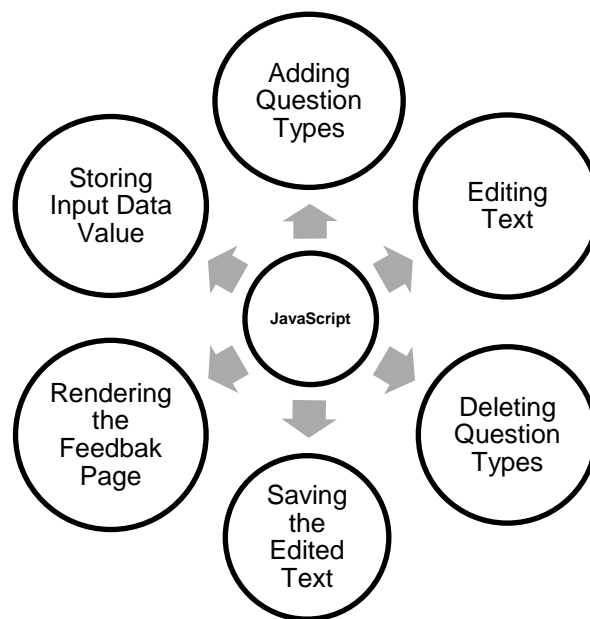


Figure  
5.  
Implem

entation area of JavaScript in the feedback application

JavaScript along with jQuery library serves a set of actions in our application like adding questionnaire, delete or edit each question, saving the edited text, rendering the preview page (Student's View) and finally deliver the answer input to the local storage.

### **Adding questions by clicking the button (Teacher's View)**

In the buttons panel section of *Teacher's View* integrated with a set of questionnaire buttons. All buttons are set in HTML markup defining a button id like, five stars rating questions in following way:

```
<button id="five-star" class="fb-button" >&#9734 rating 5</button>
```

In HTML documents we have jQuery `.click(function ()` for each question type to collaborate with JavaScript by assigning common function name.

```
/*5 star rating*/
$('#five-star').click(function () {
    insert5star($('#main'), undefined, true);
});
```

jQuery needs to be able to iterate with JavaScript in over all mounted implementation system. jQuery and JavaScript maintains a link for a function to execute each time the event is triggered. The block of following code shows basic example of how JavaScript defined for five star rating questions in external JavaScript file:

```
/*function for 5star rating choice question*/

function insert5star(elem, questionObj, editable) {
    if((typeof(questionObj) !== 'object') && editable){
        questionObj = {
            type: '5starRating',
            question: '<h3 class="first-header">Heading</h3><p>Double click on the text to edit </p>'
        }
    }
    var itemID = $(''.5starRating').length;
    elem.append(
        '<section class="5starRating" id="5star-'+itemID+'"><div data-
question>'+questionObj.question+'</div>'
        + '<form>'
        + ' <span class="star-cb-group">'
        + ' <input type="radio" id="rating5'+ itemID + '_' + 1 + '" class="fivestar"
name="rating" value="5" /><label class="rating5" for="rating5'+ itemID + '_' + 1 + '" title="Very
Satisfied"></label>'
        + .....
        + '</section>'
    );
}
```

This combination serves two purposes. First, in the HTML page we set button for adding five star rating by assigning jQuery function, `insert5star`. Second, jQuery call for that specific function in JavaScript file to represent five star rating question in HTML page according to pre-defined structure.

### Edit and delete question

To edit and delete any question we present an approach that is we think flexible enough for the user to perform the action. When the questions added in the page, they each come with their own delete button with editing capabilities.

Select

Double click on the text and write your question here.

Yes
  No
  May be

Delete

Picture 11. Edit and delete function for *select* type question

There is a delete button with every question types, which perform delete action for that specific question. The button is identified HTML with unique id, so that JavaScript function can be executed by the following way:

```
+ (editable ? '<button id="del-range" value="delete">Delete</button>' : '')
```

```
/*jQuery function for deleting current question*/
```

```
$( 'body' ).on( 'click', '#del-range', function () {
    $( this ).parent().remove();
  });
```

Instead of using any edit button we applied little tricks to edit the text. Double clicking on the text, trigger a function to enable text editing for specific question. Here, we allowed all the 'p' and 'h3' elements to be editable by assigning following codes below in the HTML document. To let the user know about editing option, all the questions, for example in Picture 11, come along with instruction: *"Double click on the text and write your question here"*.

```
/*editing questions*/
$( 'body' ).on( 'dblclick', 'p', function () {
    $( this ).attr( 'contentEditable', true );
  });

$( 'body' ).on( 'dblclick', 'h3', function () {
    $( this ).attr( 'contentEditable', true ); });
```

## Rendering the Student's View

The idea of representing form preview using JavaScript is not new. The question is how we can customize in our own work. The approach advocated in the *Student's View*, based on the JavaScript form functionalities. As mentioned earlier, we have one basic UI layout for this application and that is *Teacher's View*. We have developed the UI layout in a

strategic way, so that it can also be customized for the *Student's View* without creating separate elements for that.

In the HTML and CSS section, we defined the page markup and style for necessary elements. JavaScript does rest of the work to serve the purpose of this application. In order to render the feedback page for students, we simply defined the elements to be shown in *Student's View* by using JavaScript function with specific IDs and CSS classes. When the 'Save and Preview' button is pressed after preparing the questionnaire, it generates the evaluation form for students with given instructions by JavaScript function.

```
$(document).ready(function () {
  $('#previewr').click(function () {
    saveQuestions();
    var w = window.open("", 'win', 'width=1000,height=1000');
    var html = '<html><head>'
      + '<link rel="stylesheet" href="css/basic-style.css" />'
      + '<script src="http://code.jquery.com/jquery-1.7.1.min.js"></script>'
    + '</head><body data-editable="false">'
      + '<div class="wide-content"><header class="wrapper clearfix"><div id="banner"><div id="logo"><a'
      href="ville.html"></a></div></div></header><section id="hero"
      class="clearfix"><div class="wrapper"><h1>....</h1><p>.... </p></div></section></div>'
      + '<div id="main" class="wrapper"></div>'
      + '<button id="send-preview" style="margin-left:200px;" class="buttonlink" type="submit"
      value="send">Submit</button>'
      + '<footer class="footer">...</footer>'
      + '<script src="js/preview.js"></script>'
      + '</body></html>';
    w.document.open();
    w.document.write(html);
    w.document.close();
    return false;
  });
});
```

The block of code above performs the preview function of *Student's View*. It saves the page contents after adding and editing the questionnaire, and then renders the preview (*Student's View*) with defined markup in a new window.

## 4.6 JSON

JSON is an abbreviation of JavaScript Object Notation, which allows storing information of a Web document in an organized, easy-to-access manner. JSON syntax performs as a substitute of XML to transmit data between a server and Web application. It can be denoted as an open standard format that uses human-readable text to transfer data objects into data representation of computing systems. However, JSON is considered as a language-independent data format, although it is simulated from the JavaScript scripting language. It contains different elements to serve the purpose of use. Some of the basic elements are depicts in the table below:



Table 5. Basic elements of JSON notation and their method of use (JSON, Webopedia, 2014)

Elements	Meaning
Objects	Like object in other programming language and it begin and end with curly braces ({})
Object Members	strings and values, separated by colon (:)
Arrays	Use to hold multiple objects and values are separated by commas
Values	a string, a number, an object, an array, or the literals true, false or null.
Strings	contain characters or common backslash escapes and surrounded by double quotes

#### 4.6.1 Collecting Input Data Using JSON

To store the students feedback evaluation input data, we used JSON function to put it in the local storage. In addition to this, Web applications are able to store data locally within the user's browser using local storage without effecting Website performance. This thesis project's work basically aims to show how data can be transferred to the main server for later analysis. However, storing data using JSON can be implemented in many ways. We presented an approach that uses array method which collects all questions of the questionnaire and their input values in a correct sequence after submitting the feedback by the respondent.

Here, the `JSON.stringify()` method converts a value to JSON including the specified properties. Then all the questions are arranged in a title 'question' with comma separated value maintaining the questionnaire sequence.

```
localStorage.setItem('questions', JSON.stringify(questions));
```

The following block of code shows how JSON is implemented in the feedback application to transfer data to the local storage.

```
sendSubmission = function (questionAnswers) {
```

```

getQuestions = function () {
    return JSON.parse(localStorage.getItem('questions'));  localStorage.put('questionAnswers',
JSON.stringify(questionAnswers));
}
/*
    This gets called when preview submit button is pressed. It collects the user's answers from all of the
questions.
*/
collectQuestionAnswers = function () {
    var allQuestions = $('div[data-question]'); // it selects the heading and questions only.
    for (var i = 0; i < allQuestions.length; ++i) {
        var currentQuestion = allQuestions.eq(i);
    }
}
}

```

The JavaScript functions go through the all questions added in the feedback form to get answer inputs and pass them to the local storage using JSON. Moreover, specific ID defines the input values of each question, so that it can be identified particularly.

In the source code, functions and their purpose of use is also mention in comments. Full source code for JavaScript section can be found in Appendix 1.0 and 3.0.

#### 4.7 Adobe PhoneGap Build

Adobe PhoneGap, allows the developer to create mobile application by using Web-based programming language, whereas PhoneGap Build is a cloud-based service for compiling PhoneGap applications.

As our application is developed using browser supported programming language, we can deploy it to the Adobe PhoneGap Build system to examine the actual user experience for the mobile user. It requires an application to be packaged in a specific way so that it can merge with the system. To test the application for mobile user all we need to do is: adding the following reference in the *index.html* file and upload the .html, .css, images, .js files, etc. in a zip file.

```
<scriptsrc="phonegap.js"></script>
```

PhoneGap Build does rest of the work and generates QR code in order to download the application from mobile. We can run the application from mobile and check the outcomes of the work.

## 5 TEST EXECUTION AND OUTCOMES

The following chapter reflects the outcome of this application. The first section of this chapter focuses on the outcome for popular browsers. It also shows the outcome for mobile user using PhoneGap Build. Input data representation is reflected in the second section.

### 5.1 Project Goals

We have developed a prototype implementation of a dynamic feedback system for ViLLE e-learning environment targeting at mobile users. This prototype is developed to be implemented as a *Peer Review* for the ViLLE system. An overview of the requirements and development procedures for the feedback application is presented throughout the thesis work, which serves as a base for the project goals.

Firstly, we described how we can develop two user interfaces maintaining one base layout. Secondly, we showed how input data can be represented using local storage for later analysis in the database system. The project managed to show proper outcome for future development in the real system.

### 5.2 Application Testing

In order to evaluate the developed prototype and the process of developing the application, we have pursued the following direction to analyse the performance.

- Creating demo questionnaire
- Checking the compatibility in browsers
- Finding the outcomes for Firefox and Chrome browser
- Checking the compatibility for different sized devices using Firefox developers tool(Responsive view)
- Test outcomes from mobile using Adobe PhoneGap Build
- Test outcomes for saving input data using browser's local storage

### 5.2.1 Creating Demo Questionnaire (Teacher's View)

**VILLE**

#### Peer Review

Give your evaluation according to the questions. The peer review has been divided into few different sections. Please check all the sections before clicking the **Submit** button.

+★ rating 3   +★ rating 5   +Multiple choice   +Yes No   +Range   +Comment   +File upload

#### Workshop Seminar

Did the game development workshop seminar fulfill your reason for attending?

Yes    No    May be

Delete

#### Instructor's Expertise

On scale of 1-10, please rate the instructor's expertise for this game development workshop.

0  10

0

Delete

#### Satisfaction Level

Please indicate your overall satisfaction level in this workshop

☹️ ♂️ 😊 😄 🌟

☆ ☆ ☆ ☆ ☆

Delete

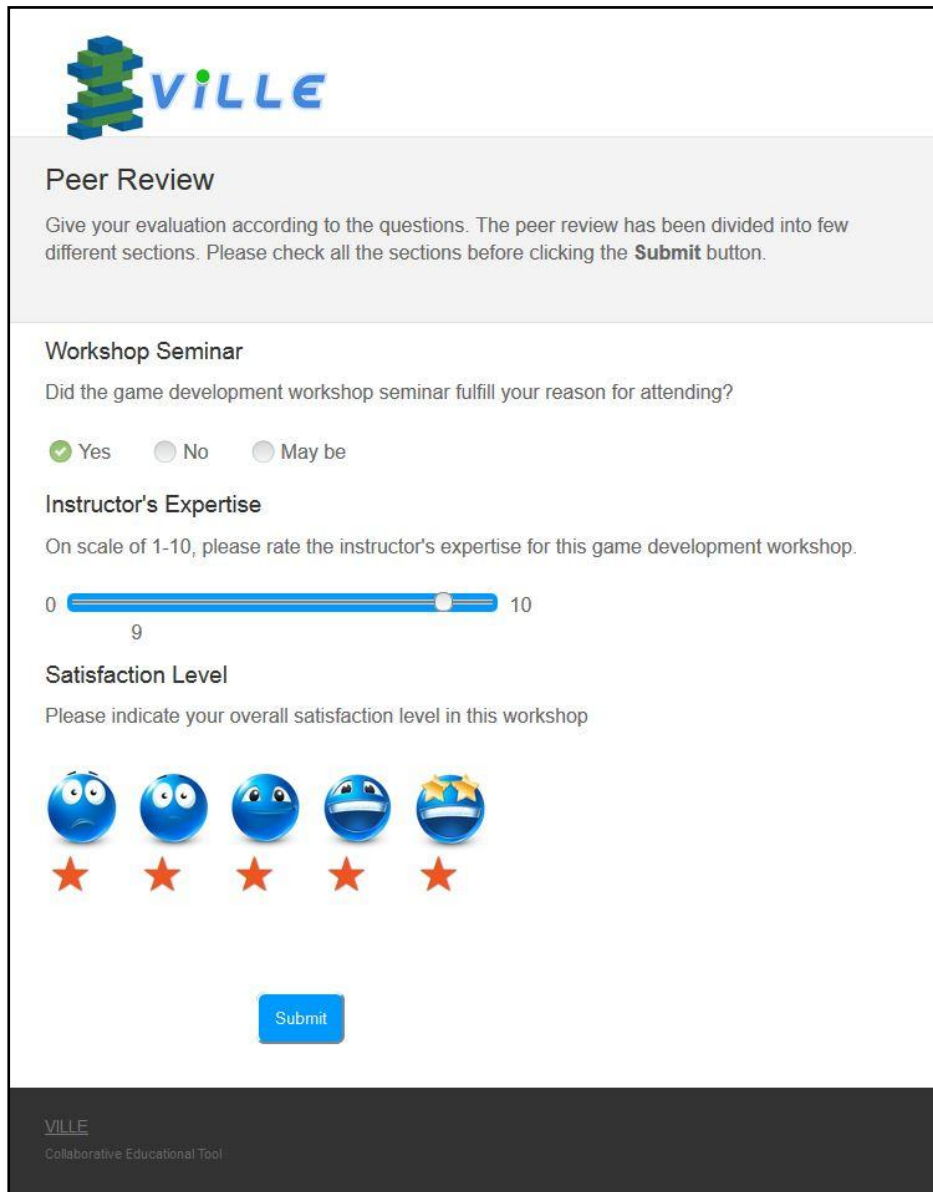
Save & preview

VILLE  
Collaborative Educational Tool

Picture 12. Demo questionnaire in the *Teacher's View* page using Firefox Web browser

Here we have added three different types of questions in the *Teacher's View* page like in Picture 12. The questions are added one by one. When the teacher clicks on 'Save and Preview' button, it saves the questionnaire and renders the *Student's View* page in a separate window.

## 5.2.2 Outcomes in Large Screen (*Student's View*)



**VILLE**

### Peer Review

Give your evaluation according to the questions. The peer review has been divided into few different sections. Please check all the sections before clicking the **Submit** button.

#### Workshop Seminar

Did the game development workshop seminar fulfill your reason for attending?

Yes    No    May be

#### Instructor's Expertise

On scale of 1-10, please rate the instructor's expertise for this game development workshop.

0  10  
9

#### Satisfaction Level

Please indicate your overall satisfaction level in this workshop

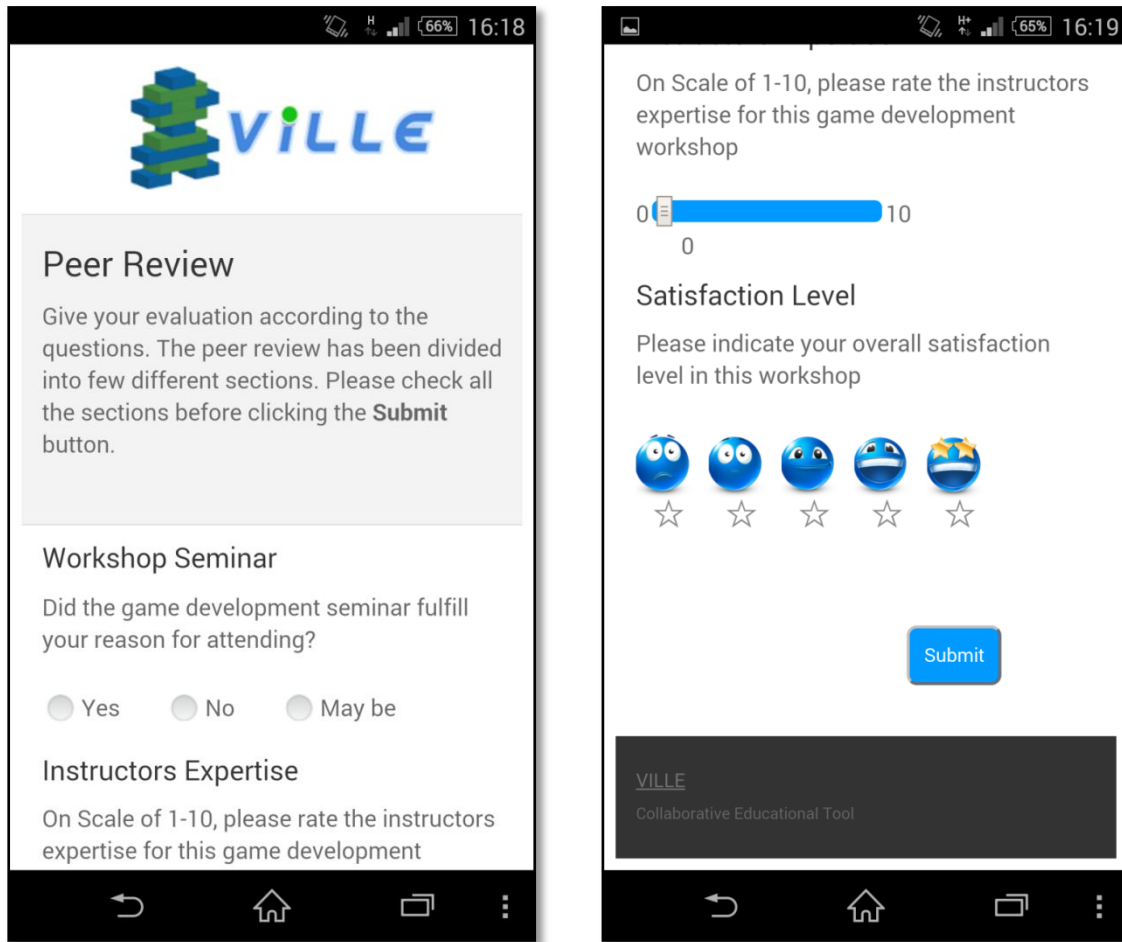
★ ★ ★ ★ ★

VILLE  
Collaborative Educational Tool

Picture 13. Demo questionnaire in the *Student's View* page using Firefox browser

The Student's View page, should be able to represent the questionnaire, set by the teacher for specific evaluation. As an example, we added three different types of questions, editing the heading and text in order to make a realistic experience. Here, Picture 13 shows the actual outcome concerning the purpose of this application. The following section also represents an evaluation made to serve the test execution for input data storing. Later in the data collection section we will see the result in local storage for these input data values.

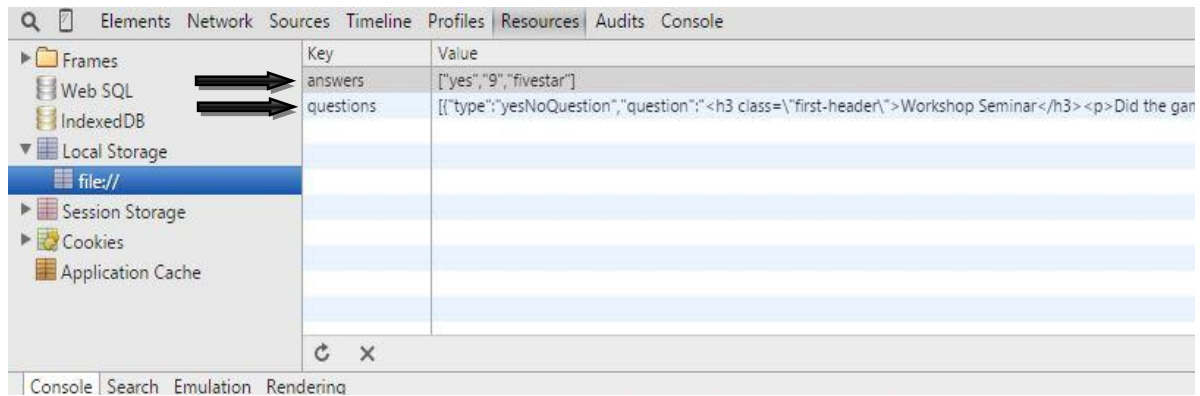
### 5.2.3 Mobile View Test Using PhoneGap



Picture 14. Results from mobile view using PhoneGap Build

The following section represents a test evaluation for the mobile user by presenting an overview of the small devices and comparing the difference with large sized devices. To perform this test we have uploaded our source file in the Adobe Phonegap Build system then installed it into an android device to see the actual outcome. The test for small devices provides a successful outcome by showing proper user experience. The result shows, all the user input elements work perfectly in the touch enable device. As the media queries are provided for different browsers to enable responsive view, it performs accordingly.

### 5.3 Validating Input Data



Picture 15. Results from the local storage in Google Chrome browser for demo questionnaire

In the earlier chapter, we mentioned that JSON is able to pass the user's input into the local storage of a browser. The following outcome shows the user inputs in an organized way, which was performed in Picture 15. All the questions and their answers value are saved in a comma separated format.

## 6 CONCLUSION

This thesis project was developed as a prototype experiment of a Web based feedback application for ViLLE e-learning environment using browser supported programming language to show an effective implementation method in the real environment providing responsive experience. The application's concept and design has a goal to show how efficiently we can serve the smart phone users. To fulfill this aim there was also a need to determine what components are required by such an application and what the possible benefits it would bring. Throughout our development procedure, we tried to include suitable elements for feedback questionnaire to achieve a concrete outcome. In order to deal with this, we have looked through many survey and feedback questionnaire models to pick the best components suitable for Web platform concentrating smart phone users. In addition to this, the questionnaire explained in Chapter 3, was developed and later tested with mobile a device.

During the application's development procedure, background knowledge about user interface design and programming language was also conducted focusing on related context of our application. The test execution also covered the area that have been focused on along with the demonstrations of two main users interface and test outcomes in the development phase.

The coding section, Chapter 4, mostly concentrated on the core functions and their implementation in the project. However, there are some other issues that could be described in order to clarify the whole concept of the coding structure. The source codes referred to in the Appendix, clarify most of the issues using comment tags that were not described in Chapter 4.

When the application satisfied all the requirements, a user study was conducted in order to assess the usability of the prototype. The results of the user testing are provided in Chapter 5 and show an overall positive user experience with possible outcomes.

The thesis describes our main concept and development procedure showing the experience with two different implementations of the application. It describes the methodology of developing a dynamic feedback application using a base page layout. Finally, it demonstrates how reviewer's input is organized to implement in the real database system for future analysis.



## 5.4 Future Work

An effective feedback always plays a vital rule in the context of performance analysis. This thesis work mainly covers the development and distribution phase of a feedback form for evaluation. Data analysis was not implemented in this section of this thesis work. This thesis provides thoughts and suggestions for further development in the database system to analyze the data. Future development could focus on some of the issues listed below:

- Generation of an automatic URL for each questionnaire to distribute the feedback form
- Development of a new interface for data representation
- Data analysis in a mathematical way
- Building an independent application for cross-platform using PhoneGap

## REFERENCES

- Bishop, I. D. (1997). Testing perceived landscape color-difference using the Internet. *Landscape and Urban Planning*, 37 (3–4): 187–196.
- Bachmann, D., & Elfrink, J. (1996). Tracking the progress of e-mail versus snail-mail. *Marketing Research*, 8(2), 31–35.
- Cummings, R., Ballantyne, C., & Fowler, L. (2000). Online student feedback surveys: Encouraging staff and student use. In *Student feedback on teaching: Reflections and projections*. Refereed proceedings of Teaching Evaluation Forum held (pp. 29-37).
- Frain, B. (2012). *Responsive Web Design with HTML5 and CSS3 (Community Experience Distilled)*. Edition. Packt Publishing.
- Hawkins, P. J. (2012). *Data Representation Synthesis*. 3.0. United States: Stanford University
- Idowu, F. M. (2006). *Online Survey System A Web-Based Tool for Creating and Administering Student Evaluations Online*.
- Kelly, M. & Marsh, J. (1999). *Going online with student evaluation of teaching*. Hong Kong: Evaluation of the Student Experience Project, City University of Hong Kong.
- Kiesler, S., & Sproull, L. (1986). Response effects in the electronic survey. *Public Opinion Quarterly*, 50, 402–413.
- Mathieu, B. (2011). *Native Cross-platform Mobile Application Development Using Voind*. 1.0. Netherlands: Delft University of Technology.
- Parker, L. (1992). Collecting data the e-mail way. *Training and Development*, 52–54.
- Stanton, J. M. (1998). An empirical assessment of data collection using the Internet. *Personnel Psychology*, 51(3), 709–726.
- Schaefer, D. R., & Dillman, D. A. (1998). Development of standard e-mail methodology: Results of an experiment. *Public Opinion Quarterly*, 62(3), 378–397.
- Sproull, L. S. (1986). Using electronic mail for data collection in organizational research. *Academy of Management Journal*, 29, 159–169.
- Tosti, D.T. (1986). *Feedback Systems*. In *Introduction to performance technology* (pp. 150-169). Washington, D.C: National Society for Performance and Instruction.
- Yun, G. W. and Trumbo, C. W. (2000), *Comparative Response to a Survey Executed by Post, E-mail, & Web Form*. *Journal of Computer-Mediated Communication*, 6: 0.

## ONLINE DOCUMENT

Applications architecture - Wikipedia, the free encyclopedia. 2014. Applications architecture - Wikipedia, the free encyclopedia. [ONLINE] Available at: [http://en.wikipedia.org/wiki/Applications\\_architecture](http://en.wikipedia.org/wiki/Applications_architecture). [Accessed 01 October 2014].

Feedback Systems. 2014. Feedback Systems. [ONLINE] Available at: [http://edWeb.sdsu.edu/people/arossett/pie/interventions/feedback\\_1.htm](http://edWeb.sdsu.edu/people/arossett/pie/interventions/feedback_1.htm). [Accessed 1 November 2014].

JavaScript - Wikipedia, the free encyclopedia. 2014. JavaScript - Wikipedia, the free encyclopedia. [ONLINE] Available at: <http://en.wikipedia.org/wiki/JavaScript>. [Accessed 27 October 2014].

jQuery Introduction. 2014. jQuery Introduction. [ONLINE] Available at: [http://www.w3schools.com/jquery/jquery\\_intro.asp](http://www.w3schools.com/jquery/jquery_intro.asp). [Accessed 05 November 2014].

jQuery Get Content and Attributes. 2014. jQuery Get Content and Attributes. [ONLINE] Available at: [http://www.w3schools.com/jquery/jquery\\_dom\\_get.asp](http://www.w3schools.com/jquery/jquery_dom_get.asp). [Accessed 05 November 2014].

Prototype - Wikipedia, the free encyclopedia. 2014. Prototype - Wikipedia, the free encyclopedia. [ONLINE] Available at: <http://en.wikipedia.org/wiki/Prototype>. [Accessed 14 December 2014].

Question Types Guide | Qualtrics. 2014. Question Types Guide | Qualtrics. [ONLINE] Available at: <http://www.qualtrics.com/university/researchsuite/basic-building/editing-questions/question-types-guide/>. [Accessed 02 November 2014].

*Responsive Web design* - Wikipedia, the free encyclopedia. [ONLINE] Available at: [http://en.wikipedia.org/wiki/Responsive\\_Web\\_design](http://en.wikipedia.org/wiki/Responsive_Web_design). [Accessed 1 November 2014].

Responsive Design in 3 Steps. 2014. Responsive Design in 3 Steps. [ONLINE] Available at: <http://Webdesignerwall.com/tutorials/responsive-design-in-3-steps>. [Accessed 27 October 2014].

Software development process - Wikipedia, the free encyclopedia. 2014. Software development process - Wikipedia, the free encyclopedia. [ONLINE] Available at: [http://en.wikipedia.org/wiki/Software\\_development\\_process#Prototyping](http://en.wikipedia.org/wiki/Software_development_process#Prototyping). [Accessed 02 November 2014].

Semiotics - Wikipedia, the free encyclopedia. 2014. Semiotics - Wikipedia, the free encyclopedia. [ONLINE] Available at: <http://en.wikipedia.org/wiki/Semiotics>. [Accessed 1 November 2014].

*Software development process* - Wikipedia, the free encyclopedia. [ONLINE] Available at: [http://en.wikipedia.org/wiki/Software\\_development\\_process#Prototyping](http://en.wikipedia.org/wiki/Software_development_process#Prototyping). [Accessed 1 November 2014].

Sample Survey Question Formats. 2014. Sample Survey Question Formats. [ONLINE] Available at: <http://www.custominsight.com/survey-question-types.asp>. [Accessed 01 November 2014].

Use CSS media queries for responsiveness — Web Fundamentals. [ONLINE] Available at: <https://developers.google.com/Web/fundamentals/layouts/rwd-fundamentals/use-media-queries?hl=en>. [Accessed 11 November 2014]

User interface design - Wikipedia, the free encyclopedia. 2014. User interface design - Wikipedia, the free encyclopedia. [ONLINE] Available at: [http://en.wikipedia.org/wiki/User\\_interface\\_design](http://en.wikipedia.org/wiki/User_interface_design). [Accessed 01 October 2014].

Use CSS media queries for responsiveness — Web Fundamentals. 2014. Use CSS media queries for responsiveness — Web Fundamentals. [ONLINE] Available at: <https://developers.google.com/Web/fundamentals/layouts/rwd-fundamentals/use-media-queries?hl=en>. [Accessed 25 October 2014].

VILLE - Collaborative Education Tool. 2014. VILLE - Collaborative Education Tool. [ONLINE] Available at: <https://ville.cs.utu.fi/>. [Accessed 01 December 2014].

W3Schools Online Web Tutorials. 2014. W3Schools Online Web Tutorials. [ONLINE] Available at: <http://www.w3schools.com/>. [Accessed 03 November 2014].

World Wide Web Consortium (W3C). 2014. World Wide Web Consortium (W3C). [ONLINE] Available at: <http://www.w3.org/>. [Accessed 03 November 2014].

What is JSON? - A Word Definition From the Webopedia Computer Dictionary. 2014. What is JSON? - A Word Definition From the Webopedia Computer Dictionary. [ONLINE] Available at: <http://www.webopedia.com/TERM/J/JSON.html>. [Accessed 01 December 2014]

Word Definition From the Webopedia Computer Dictionary. [ONLINE] Available at: <http://www.Webopedia.com/TERM/J/JSON.html>. [Accessed 01 December 2014].

## APPENDICES

### Appendix 1.0 Source code for the HTML file (index.html)

```

<!DOCTYPE html>
<html class="no-js">
<head>
<meta charset="utf-8">
<title>Ville peer review </title>

<!-- Mobile viewport -->
<meta name="viewport" content="width=device-width; initial-scale=1.0">
<link rel="shortcut icon" href="images/favicon.ico" type="image/x-icon" />

<!-- CSS-->
<link rel="stylesheet" href="css/basic-style.css">
<!-- end CSS-->
<!-- jQuery -->
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1/jquery.min.js"></script>
<!-- JS-->
<script src="js/libs/modernizr-2.6.2.min.js"></script>
<script src="http://code.jquery.com/jquery-1.7.1.min.js"></script>
<script src="js/preview.js"></script>

<script>
function saveQuestions() {
// Loop through the questions and build JSONs.
var questions = [];
var questionElements = $('section.yesNoQuestion, section.3starRating,
section.multipleChoiceQuestion, section.5starRating, section.rangeQuestion, section.commentbox');

for (var i = 0; i < questionElements.length; ++i) {
var current = questionElements.eq(i);
var item = {
type: current.attr('class'),
question: current.find('[data-question]').eq(0).html()
}
questions.push(item);
}
localStorage.setItem('questions', JSON.stringify(questions));
}

/*for preview page*/
$(document).ready(function () {
$('#previewr').click(function () {
saveQuestions();

var w = window.open("", 'win', 'width=1000,height=1000');
var html = '<html><head>'
+ '<link rel="stylesheet" href="css/basic-style.css" />'
+ '<script src="http://code.jquery.com/jquery-1.7.1.min.js"></script>'

//+ '<script>$(document).ready(function(e) {$("#main").append("hello.");})</script>'
+ '</head><body data-editable="false">'
//+ $('#wrapper').html()
+ '<div class="wide-content"><header class="wrapper clearfix"><div id="banner"><div id="logo"><a
href="ville.html"></a></div></div></header><section id="hero"
class="clearfix"><div class="wrapper"><h1>Peer Review</h1><p>Give your evaluation according to the scale
of 1-5 star.The peer review has been splitted into few different sections.Star 1 shows the least points and star

```

5 is the greatest. Please check all the section before clicking the <strong>Submit</strong> button.

```

</p></div></section></div>'
+ '<div id="main" class="wrapper"></div>'
+ '<button id="send-preview" style="margin-left:200px;" class="buttonlink" type="submit"
value="send">Submit</button>'
+ '<footer class="footer"><div id="colophon" class="wrapper clearfix"><a href="http://ville.cs.utu.fi/"
target="_blank" title="Learn more about ville" style="color:#777;">VILLE</a></div><div id="attribution"
class="wrapper clearfix" style="color:#666; font-size:11px;">Collaborative Educational Tool </div><!--end
attribution--></footer>'

+ '<script src="js/preview.js"><\script>'
+ '</body></html>';
w.document.open();
w.document.write(html);
w.document.close();

return false;
});
/*3 star rating*/
$('#three-star').click(function () {
insert3star($('#main'), undefined, true);
});
/*5 star rating*/
$('#five-star').click(function () {
insert5star($('#main'), undefined, true);
});

/*multiple choice*/
$('#check-type').click(function () {
insertMultipleChoice($('#main'), undefined, true);
});

$('#slide').click(function (e) { insertSlider($('#main'),undefined, true); });

$('body').on('click', '#del-range', function () {
$(this).parent().remove();
});

/*editing questions*/
$('body').on('dblclick', 'p', function () {
$(this).attr('contentEditable', true);
});

/*yes No question*/
$('#radio-button').click(function () {

insertYesNo($('#main'), undefined, true /*maybe use a variable here that tells if the questions can be
edited or not.*/);
});

/*comment box*/
$('#comment-box').click(function(){

insertcommentbox($('#main'), undefined, true);
});

});

</script>
<!-- end JS-->

```

```

<style type="text/css">
</style>
</head>

<body data-editable="true">
<div id="wrapper">
<!-- header area -->
  <header class="wrapper clearfix">

    <div id="banner">
      <div id="logo"><a href="ville.html"></a></div>
    </div>

  </header><!-- end header -->

<!-- header-->

  <section id="hero" class="clearfix">

    <div class="wrapper">
      <h1>Peer Review</h1>
      <p> Give your evaluation accroding to the
scale of 1-5 star. The peer review has been splited into few different sections.
      Star 1 shows the least points and
star 5 is the greatest. Please check all the section before clicking the
      <strong>Submit</strong> button.
    </p>
    </div>

  </section><!--end header-->

  <section style="text-align:center;" id="content" class="wide-content">
    <button id="three-star" class="fb-button" > &#43; &#9734 rating 3</button>
    <button id="five-star" class="fb-button" > &#43; &#9734 rating 5</button>
    <button id="check-type" class="fb-button" value="Multiple choice"> &#43; Multiple choice</button>
    <button id="radio-button" class="fb-button" > &#43; Yes No</button>
    <button id="slide" class="fb-button" > &#43; Range</button>
    <button id="comment-box" class="fb-button" > &#43; Comment</button>
    <button id="file-upload" class="fb-button" > &#43; File upload</button>
  </section>

<!-- main content area -->
<div id="main" class="wrapper">

</div><!-- #end div #main .wrapper -->

  <div class="grid_4">

    <button style="margin-left:200px;" id="previewr" class="fb-button" class="wide-content">Save &
preview</button>
  </div><br></br>
  </form>

<!-- footer area -->
  <footer class="footer">

```

```

        <div id="colophon" class="wrapper clearfix">
            <a href="http://ville.cs.utu.fi/" target="_blank" title="Learn more about ville"
style="color:#777;">VILLE</a>
        </div>
        <div id="attribution" class="wrapper clearfix" style="color:#666; font-size:11px;">Collaborative
Educational Tool </div><!--end attribution-->

    </footer><!-- #end footer area -->
</div>
</body>
</html>

```

## Appendix 2.0 Source code for the CSS file (basic-style.css)

```

/*
Ville Peer Review Project by ICT-Portti
BASE (MOBILE) SIZE
    These are the mobile styles. It's what people see on their phones.
*/
body{
    color:#666;
    font-family:'Ubuntu', Arial, Helvetica, sans-serif;
    font-size:1em;
    line-height:1.4em;
    font-weight:normal;
}
h1, h2, h3, h4, h5, h6{
    font-weight:normal;
    font-family:'Droid Serif', Arial, Helvetica, sans-serif;
    line-height:1.5em;
    margin:.45em 0;
    padding:0;
}

/* links */
a,
a:visited,
a:active,
a:hover{color:#0099ff;}
a:hover{ text-decoration:none;}

/* Box sizing. Read about it here: http://www.w3schools.com/cssref/css3_pr_box-sizing.asp */
*{
    box-sizing:border-box;
    -moz-box-sizing:border-box;}

/* structure */
.wrapper{
    width: 92%;
    margin: 0 auto;
}
header{
    padding:15px 0;
}
#banner{

```



```

        text-align:center;
    }

    #hero,
    #page-header{
        background:#f3f3f3;
        border-top:1px solid #e2e2e2;
        border-bottom:1px solid #e2e2e2;
        padding:20px 0;
    }
    #hero h1{
        line-height:1.2em;
        margin-top:0px;
        margin-bottom:10px;}

    .flexslider{
        display:none;
    }

    #content {
        margin:50px 0;
    }

    aside {
        margin:40px 0;
    }

    p{ margin:0 0 1.5em;}

    /* RESPONSIVE IMAGES */
    img{ max-width:100%; height:auto;}

    .star-cb-group {
        /* remove inline-block whitespace */
        font-size: 0;
        /* flip the order so we can use the + and ~ combinators */
        unicode-bidi: bidi-override;
        direction: rtl;
        /* the hidden clearer */
        /* this is gross, We threw this in to override the starred
        buttons when hovering. */
    }
    .star-cb-group * {
        font-size: 40px;
    }
    .star-cb-group > [type*="radio"] {
        display: none;
    }
    .star-cb-group > [type*="radio"] + label {
        /* only enough room for the star */
        display: inline-block;
        height: 30px;
        overflow: hidden;
        padding: 80px 37px;
        text-indent: 9999px;
        white-space: nowrap;
        width: 30px;
    }
    .star-cb-group > [type*="radio"] + label:before {
        display: inline-block;

```

```

text-indent: -9999px;
content: '\2606';
/* WHITE STAR */
color: #888;
}
.star-cb-group > [type*="radio"]:checked ~ label:before,
.star-cb-group > [type*="radio"] + label:hover ~ label:before,
.star-cb-group > [type*="radio"] + label:hover:before {
content: '\2605';
/* Color STAR */
color: #e52;
text-shadow: 0 0 1px #333;
}
.star-cb-group > .star-cb-clear[type*="radio"] + label {
text-indent: -9999px;
width: 100%;
margin-left: -.5em;
}
.star-cb-group > .star-cb-clear[type*="radio"] + label:before {
width: 100%;
height: 100%;
}
.star-cb-group:hover > [type*="radio"] + label:before {
content: '\2606';
/* WHITE STAR */
color: #888;
text-shadow: none;
}
.star-cb-group:hover > [type*="radio"] + label:hover ~ label:before, .star-cb-group:hover > [type*="radio"] +
label:hover:before {
content: '\2605';
/* Color STAR */
color: #e52;
text-shadow: 0 0 1px #333;
}

label.rating5 {
background: url("../images/5.png") no-repeat scroll -7px 0 / 100% auto rgba(0, 0, 0, 0);
}

label.rating4 {
background: url("../images/4.png") no-repeat scroll -7px 0 / 100% auto rgba(0, 0, 0, 0);
}
label.rating3 {
background: url("../images/3.png") no-repeat scroll -7px 0 / 100% auto rgba(0, 0, 0, 0);
}
label.rating2 {
background: url("../images/2.png") no-repeat scroll -7px 0 / 100% auto rgba(0, 0, 0, 0);
}
label.rating1 {
background: url("../images/1.png") no-repeat scroll -7px 0 / 100% auto rgba(0, 0, 0, 0);
}
@media all and (max-width: 480px) {
.star-cb-group > [type*="radio"] + label {
/* only enough room for the star */
display: inline-block; height: 20px; overflow: hidden; padding: 50px 15px; text-indent: 9999px; white-space:
nowrap; width: 50px; }
.star-cb-group * { font-size: 25px; }

.slider-rating /* For the NPS Slider */
input[type="range"]{

```

```

background-color:#0099ff;
width: 50%;
height: 15px;
-Webkit-appearance:none;
border-radius: 5px;
border-style:solid;
border-color:#0099ff;

}
}

@media all and (max-width: 320px) {
.star-cb-group > [type*="radio"] + label {
/* only enough room for the star */ display: inline-block;border: 0; height: 25px; overflow: hidden; padding:
45px 23px; text-indent:9999px; white-space: nowrap; width: 50px; }
.star-cb-group * { font-size: 17px; }

.slider-rating
input[type="range"]{
background-color:#0099ff;
width: 50%;
height: 15px;
-Webkit-appearance:none;
border-radius: 5px;
border-style:solid;
border-color:#0099ff;

}
}

/* -----This is for yes no question-----http://www.csscheckbox.com/checkbox/7569/bubbly-
green-bomberman-checkbox/----- */
.css-checkbox {
display:none;
}

input[type=radio].css-checkbox + label.css-
label {
padding-left:24px;
padding-right:30px;
height:19px;
display:inline-block;
line-height:19px;
background-repeat:no-repeat;
background-position: 0 0;
font-size:inherit;
vertical-align:middle;
cursor:pointer;
}

input[type=radio].css-checkbox:checked +
label.css-label {
background-position: 0 -19px;
}
label.css-label {
background-image:url(..images/checkbox.png);
}
}

```

```

.css-checkbox{
    border-style:none;
}
.css-checkbox {
    text-align: center;
}
/* -----This is for yes no question----- */
/* Style of the table in rating section */
.rating{
    border-style:none;
}
.rating{
    text-align:center;
}
}
/* Style of the table in rating section */

/* Slider css codes */
.slider-rating
    input[type="range"]{
        background-color:#0099ff;
        width: 50%;
        height: 15px;
        -Webkit-appearance:none;
        border-radius: 5px;
        border-style:solid;
        border-color:#0099ff;
    }

}

/* End of Slider css codes */

/* ----- Select Question css style from
http://www.csscheckbox.com/checkbox/7569/bubbly-green-bombman-checkbox/ */

/* ----- That select question css codes ----- */
.multipleChoiceQuestion label {
    display: inline-block;
    cursor: pointer;
    position: relative;
    padding-left: 25px;
    margin-right: 15px;
    font-size: 17px;
}

.multipleChoiceQuestion label:before {
    content: "";
    display: inline-block;

    width: 16px;
    height: 16px;

    margin-right: 10px;
    position: absolute;
    left: 0;
    bottom: 1px;

    box-shadow: inset 0px 2px 3px 0px rgba(0, 0, 0, .3), 0px 1px 0px 0px rgba(255, 255, 255, .8);
}

```

```

.multipleChoiceQuestion input[type=checkbox] {
  display: none;
}

.multipleChoiceQuestion label:before {
  border-radius: 3px;
}

.multipleChoiceQuestion input[type=checkbox]:checked + label:before {
  content: "\2713";
  text-shadow: 1px 1px 1px rgba(0, 0, 0, .2);
  font-size: 15px;
  color: #009900;
  text-align: center;
  line-height: 15px;
}

/* -----End of Select question css codes----- */
/*MAIN MENU*/
.menu-toggle{
  display:block;
  padding:10px;
  margin:20px 0 0;
  background:#666;
  color:#fff;
  cursor:pointer;
  text-transform:uppercase;
  font-size:20px;
}

/*SECONDARY MENU*/
.fb-button{display:inline-block;margin:0;padding:.563rem .844rem;border:0
none;background:#16a085;color:#fff;text-align:center;text-decoration:none;font-size:12px;line-
height:1.5;cursor:pointer;border-radius:.125rem;border:thin solid #19b394;border-bottom:2px solid #16a085}
.fb-button[disabled]{background:#ddd !important;border:thin solid #ccc;color:#777 !important;text-
shadow:none !important;-ms-
filter:"progid:DXImageTransform.Microsoft.Alpha(Opacity=65)";opacity:.65;cursor:default}

#secondary-navigation{
  margin-bottom:60px;
}
#secondary-navigation ul{
  margin:0;
  padding:0;
}
#secondary-navigation ul li a{
  background:#E6E6E6;
  display:block;
  margin:5px 0;
  padding:10px;
  text-decoration:none;
}
#secondary-navigation ul li a:hover,
#secondary-navigation ul li.current a{
  background:#0099ff;
  color:#fff;
}

/*SPACE GRID ELEMENTS VERTICALLY, SINCE THEY ARE ONE UNDER ANOTHER SO FAR*/

```

```

.grid_4
{
    margin-bottom:40px;
}

/*FOOTER*/
footer{
    clear:both;
    font-size:80%;
    padding:20px 0;
}
footer ul{
    margin:0;
    padding:0;
}

/*colors and backgrounds*/
body{
    background:#fff;
}
h1, h2, h3, h4, h5, h6{
    color:#333;
}

footer{
    background:#333;
    color:#ccc;
}
footer h1, footer h2, footer h3, footer h4{
    color:#CCC;
    margin-bottom:10px;
}
footer ul{
    margin:0 0 0 8%;
}

.buttonlink{
    background:#0099ff;
    border-radius:7px;
    color:#fff;
    display:block;
    float:left;
    margin:10px 15px 35px 120px;
    padding:10px;
    text-decoration:none;
    text-align:center;
}

.buttonlink:hover{
    background:#8dbc01;
}

.greenelement{
    background:#5ec79e;
    color:#fff;
}

.violetelement{
    background:#887dc2;
    color:#fff;
}

```

```

/* Contain floats: h5bp.com/q */
.clearfix:before, .clearfix:after { content: ""; display: table; }
.clearfix:after { clear: both; }
.clearfix { zoom: 1; }

.rightfloat{
    float:right;
}
.leftfloat{
    float:left;
}

/*
LARGER MOBILE DEVICES
This is for mobile devices with a bit larger screens.
*/
@media only screen and (min-width: 481px) {
#banner{
    float:left;
    text-align:left;
    margin-bottom:-20px;/*this depends on the height of the logo*/
}
.menu-toggle{/*make menu float right, instead of sitting under the logo*/
    margin-top:10px; /*this depends on the height of the logo*/
    float:right;
}
}

/*
TABLET & SMALLER LAPTOPS
The average viewing window and preferred media query for those is 768px.
But we think that some more breathing space is good:)
*/
@media only screen and (min-width: 920px) {
.wrapper{
    max-width: 1200px;
    margin: .75em auto;
}
#banner{
    float:left;
    text-align:left;
    margin-bottom:0px;
}
header{
    padding:0;
}
#content {
    float:left;
    width:65%;
}
#content.wide-content{
    float:none;
    width:100%;
}
}

```

```

.flexslider{
display:block;
/*demo 1 slider theme*/
margin: 0 0 60px;
background: #fff;
border: 4px solid #fff;
-Webkit-border-radius: 4px;
-moz-border-radius: 4px;
-o-border-radius: 4px;
border-radius: 4px;
box-shadow: 0 1px 4px rgba(0,0,0,.2);
-Webkit-box-shadow: 0 1px 4px rgba(0,0,0,.2);
-moz-box-shadow: 0 1px 4px rgba(0,0,0,.2);
-o-box-shadow: 0 1px 4px rgba(0,0,0,.2);
}

aside {
float:right;
width:30%;
}

/** MAIN MENU - ESSENTIAL STYLES **/

/*GRID*/

.grid_4 { width: 30%; }
.grid_12 {
margin-left: 1.66666666666667%;
margin-right: 1.66666666666667%;
float: left;
display: block;
}

.alpha{margin-left:0px;}
.omega{margin-right:0px;}
.rightfloat{float:right;}
}

/*
DESKTOP
This is the average viewing window. So Desktops, Laptops, and
in general anyone not viewing on a mobile device.
*/
@media only screen and (min-width: 1030px) {
}

/*
LARGE VIEWING SIZE
This is for the larger monitors and possibly full screen viewers.
*/
@media only screen and (min-width: 1240px) {
}

/*
RETINA (2x RESOLUTION DEVICES)

```



This applies to the retina iPhone (4s) and iPad (2,3) along with other displays with a 2x resolution.

```

*/
@media only screen and (-Webkit-min-device-pixel-ratio: 1.5),
  only screen and (min--moz-device-pixel-ratio: 1.5),
  only screen and (min-device-pixel-ratio: 1.5) {

}

/*
IPHONE 5 MEDIA QUERY
iPhone 5 or iPod Touch 5th generation styles (you can include your own file if you want)
*/
@media (device-height: 568px) and (-Webkit-min-device-pixel-ratio: 2) {

}

/*
PRINT STYLESHEET
*/
@media print {
  * { background: transparent !important; color: black !important; text-shadow: none !important; filter:none !important; -ms-filter: none !important; } /* Black prints faster: h5bp.com/s */
  a, a:visited { text-decoration: underline; }
  a[href]:after { content: " (" attr(href) ")"; }
  abbr[title]:after { content: " (" attr(title) ")"; }
  .ir a:after, a[href^="javascript:"]:after, a[href^="#"]:after { content: ""; } /* Don't show links for images, or javascript/internal links */
  pre, blockquote { border: 1px solid #999; page-break-inside: avoid; }
  thead { display: table-header-group; } /* h5bp.com/t */
  tr, img { page-break-inside: avoid; }
  img { max-width: 100% !important; }
  @page { margin: 0.5cm; }
  p, h2, h3 { orphans: 3; widows: 3; }
  h2, h3 { page-break-after: avoid; }
}

```

### Appendix 3.0 Source code for JavaScript file (preview.js)

```

/*
  Ville team implements the final version of this function.

  @return array of json elements with attributes:
    type      Type of the question
    questionBody  html content that shows the question content, headers, etc. Does not contain form
elements, only the question.
*/
/*for multiple checkbox item*/
$(document).ready(function(){

  $('.to-uncheck').change(function(){
    $('.check').prop('checked', false);
  });

  $('.check').change(function(){
    $('.to-uncheck').prop('checked', false);

```

```

    });
});

getQuestions = function () {
    return JSON.parse(localStorage.getItem('questions'));
}

/*
    Ville team implements the final version of this function.
*/
sendSubmission = function (questionAnswers) {
    localStorage.put('questionAnswers', JSON.stringify(questionAnswers));
}
/*
    This gets called when preview submit button is pressed. It collects the user's answers from all of the
    questions.
*/
collectQuestionAnswers = function () {
    var allQuestions = $('div[data-question]'); // it selects the heading and questions only.
    for (var i = 0; i < allQuestions.length; ++i) {
        var currentQuestion = allQuestions.eq(i);
        // TODO create a answer collector for each question type:
        // getAnswerForYesNoQuestion(questionID)
    }
}

function sliderChangeHandler() {
    var a = $(this).parent().find('.currentValue').html($(this).val());
}

/*slider*/
function insertSlider(parent,questionObj, editable) {
    if((typeof(questionObj)!== 'object') && editable){
        questionObj = {
            type: 'rangeQuestion',
            question:'<h3 class="first-header">Range</h3><p>Double click on the text and write your question
            here</p>'
        }
    }
    var newSlider = $(
        '<section id="for-slide" class ="rangeQuestion">' + '<div data-question>' + questionObj.question +
        '</div>' + '<div class="slider-rating">' +
        '<span>0</span><input class="points" type="range" min="0" max="10" value="0" /><span>10</span>'
    +
        '<article class="currentValue" style="margin-left:10%;">0</article>' +
        '</div>' + (editable ? '<button id="del-range" value="delete">Delete</button>' : "") + '</section>');

    newSlider.find('.points').change(sliderChangeHandler);
    parent.append(newSlider);
}

/*function for yes no questions*/
function insertYesNo(elem, questionObj, editable) {
    if ((typeof (questionObj) !== 'object') && editable) {
        questionObj = {
            type: 'yesNoQuestion',

```

```

        question: '<h3 class="first-header">Select</h3><p>Double click on the text and write your
question here</p>'
    }
}

var itemID = $(' .yesNoQuestion').length;
elem.append(
    '<section class="yesNoQuestion" id="yesNoDiv-' + itemID + '"><div data-question>' +
questionObj.question + '</div>'
    + '    <form>'
    + '        <table><tr><td><input type="radio" name="radiog_dark" value="1" id="radio' + itemID
+ '_' + 6 + '" class="yes css-checkbox" />'
    + '        <label for="radio' + itemID + '_' + 6 + '" class="css-label radGroup2">Yes</label>'
    + '        </td><td><input type="radio" name="radiog_dark" value="2" id="radio' + itemID + '_' +
7 + '" class="no css-checkbox"><label for="radio' + itemID + '_' + 7 + '" class="css-label
radGroup2">No</label>'
    + '        </td><td><input type="radio" name="radiog_dark" value="3" id="radio' + itemID + '_' +
8 + '" class="maybe css-checkbox"><label for="radio' + itemID + '_' + 8 + '" class="css-label
radGroup2">May be</label>'
    + '        </td></tr></table>'
    + '    </form>' + (editable ? '<button id="del-range" value="delete">Delete</button>' : '') +
'</section>'
);
}

/*function for inserting comment box*/

function insertcommentbox(elem, questionObj,editable){

    if ((typeof (questionObj) !== 'object') && editable) {
        questionObj = {
            type: 'commentbox',
            question: '<h3 class="first-header">comments</h3><p>Double click on the text and write your
question here</p>'
        }
    }
    $(' .commentbox').length;
    elem.append('<section class="commentbox">'
    + '<div data-question>' + questionObj.question + '</div>'
    + '<form>'
    + '<textarea id="message" rows="4" cols="8" placeholder="Write if you have any comments...">'
    + '</textarea>'
    + '</form>'
    + (editable ? '<button id="del-range" value="delete">Delete</button>' : '')
    + '</section>'
);
}

/*function for 3star rating choice question*/
function insert3star(elem, questionObj, editable) {
    if((typeof(questionObj) !== 'object') && editable){
        questionObj = {
            type: '3starRating',
            question: '<h3 class="first-header">3star rating</h3><p>Double click on the text and write your
question here</p>'
        }
    }
    var itemID = $(' .3starRating').length;
    elem.append(

```

```

        '<section class="3starRating" id="3star-' + itemID + "'><div data-
question>' + questionObj.question + '</div>'
        + '<form id="test">'
        + '<span class="star-cb-group">'
        + '<input type="radio" id="rating3' + itemID + '_' + 5 + '" class="threestar"
name="rating" value="5" /><label class="rating5" for="rating3' + itemID + '_' + 5 + '" title="Very
Satisfied"></label>'
        + '<input type="radio" id="rating3' + itemID + '_' + 6 + '" class="twostar"
name="rating" value="3" /><label for="rating3' + itemID + '_' + 6 + '" class="rating3" title="Average"></label>'
        + '<input type="radio" id="rating3' + itemID + '_' + 7 + '" class="onestar"
name="rating" value="1" /><label for="rating3' + itemID + '_' + 7 + '" class="rating1" title="Very
Dissatisfied"></label>'
        + '</span>'
        + '</form>'
        + (editable ? '<button id="del-range" value="delete">Delete</button>' : '')
    + '</section>'
    );
}
/*function for 5star rating choice question*/
function insert5star(elem, questionObj, editable) {
    if((typeof(questionObj) !== 'object') && editable){
        questionObj = {
            type: '5starRating',
            question: '<h3 class="first-header">Group Work</h3><p>Double click on the text and write your
question here</p>'
        }
    }
    var itemID = $(''.5starRating').length;
    elem.append(
        '<section class="5starRating" id="5star-' + itemID + "'><div data-
question>' + questionObj.question + '</div>'
        + '<form>'
        + '<span class="star-cb-group">'
        + '<input type="radio" id="rating5' + itemID + '_' + 1 + '" class="fivestar"
name="rating" value="5" /><label class="rating5" for="rating5' + itemID + '_' + 1 + '" title="Very
Satisfied"></label>'
        + '<input type="radio" id="rating5' + itemID + '_' + 2 + '" class="fourstar"
name="rating" value="4" /><label class="rating4" for="rating5' + itemID + '_' + 2 + '" title="Fairly
Satisfied"></label>'
        + '<input type="radio" id="rating5' + itemID + '_' + 3 + '" class="threestar"
name="rating" value="3" /><label for="rating5' + itemID + '_' + 3 + '" class="rating3" title="Average"></label>'
        + '<input type="radio" id="rating5' + itemID + '_' + 4 + '" class="twostar"
name="rating" value="2" /><label for="rating5' + itemID + '_' + 4 + '" class="rating2" title="Fairly
Dissatisfied"></label>'
        + '<input type="radio" id="rating5' + itemID + '_' + 5 + '" class="onestar"
name="rating" value="1" /><label for="rating5' + itemID + '_' + 5 + '" class="rating1" title="Very
Dissatisfied"></label>'
        + '</span>'
        + '</form>'
        + (editable ? '<button id="del-range" value="delete">Delete</button>' : '')
    + '</section>'
    );
}

/*function for multiple choice question*/
function insertMultipleChoice(elem, questionObj, editable) {
    if((typeof(questionObj) !== 'object') && editable){
        questionObj = {
            type: 'multipleChoiceQuestion',

```

```

        question: '<h3 class="first-header">Multiple Choice</h3><p>Double click on the text and write your
question here</p>'
    }
}
var itemID = $(' .multipleChoiceQuestion').length;
elem.append(
    '<section class="multipleChoiceQuestion" id="multipleChoice-'+itemID+'"><div data-
question>'+questionObj.question+'</div>'

        + '<form>'
            + '<input id="check' + itemID + '_' + 1 + '" type="checkbox" name="check1"
value="check1" class="check input-checkboxitem"> '
            + '<label for="check' + itemID + '_' + 1 + '">Materials</label><br> <br>'

            + '<input id="check' + itemID + '_' + 2 + '" type="checkbox" name="check2"
value="check2" class="check input-checkboxitem"> '
            + '<label for="check' + itemID + '_' + 2 + '">Exam Method</label><br><br> '

            + '<input id="check3' + itemID + '_' + 3 + '" type="checkbox" name="check3"
value="check3" class="check input-checkboxitem"> '
            + '<label for="check3' + itemID + '_' + 3 + '">Learning Method</label><br><br> '

            + '<input id="check4' + itemID + '_' + 4 + '" type="checkbox" name="check4"
value="check4" class="check input-checkboxitem">'
            + '<label for="check4' + itemID + '_' + 4 + '">Evaluation</label><br><br> '

            + '<input id="check5' + itemID + '_' + 5 + '" type="checkbox" name="check5"
value="check5" class="check input-checkboxitem">'
            + '<label for="check5' + itemID + '_' + 5 + '">User Friendliness</label><br><br>
,

            + '<input id="check6' + itemID + '_' + 6 + '" type="checkbox" name="check6"
value="check6" class="to-uncheck input-checkboxitem">'
            + '<label for="check6' + itemID + '_' + 6 + '">None</label>'
        + '</form>'
        + (editable ? '<button id="del-range" value="delete">Delete</button>' : '')

    + '</section>'
);
}

```

```

function saveAnswers(){
    var answers = [];
    var answerElements = $('section.yesNoQuestion, section.3starRating, section.5starRating,
section.multipleChoiceQuestion, section.rangeQuestion, section.commentbox');

    for(var i=0; i<answerElements.length; ++i){
        var current = answerElements.eq(i);
        var quesans = current.attr('class');

        switch(quesans){
            case 'yesNoQuestion':
                var yes = current.find('input.yes');
                var no = current.find('input.no');
                var maybe = current.find('input.maybe');
                var answer = null;
                if (yes[0].checked)answer = 'yes';
                if (no[0].checked)answer = 'no';
                if (maybe[0].checked)answer = 'maybe';

```

```

    break;

    case '3starRating':
    var onestar = current.find('input.onestar');
    var twostar = current.find('input.twostar');
    var threestar = current.find('input.threestar');
    var answer = null;
    if(onestar[0].checked) answer = 'onestar';
    if(twostar[0].checked) answer = 'twostar';
    if(threestar[0].checked) answer = 'threestar';
    break;

    case '5starRating':
    var onestar = current.find('input.onestar');
    var twostar = current.find('input.twostar');
    var threestar = current.find('input.threestar');
    var fourstar = current.find('input.fourstar');
    var fivestar = current.find('input.fivestar');
    var answer = null;
    if(onestar[0].checked) answer = 'onestar';
    if(twostar[0].checked) answer = 'twostar';
    if(threestar[0].checked) answer = 'threestar';
    if(fourstar[0].checked) answer = 'fourstar';
    if(fivestar[0].checked) answer = 'fivestar';
    break;

    case 'multipleChoiceQuestion':
    var answer = current.find('input:checked').map(function(i,el){return el.name;}).get();
    break;

    case 'rangeQuestion':
    var answer = current.find('.points').val();
    break;

    case 'commentbox':
    var answer = current.find('#message').val();

    }
    answers.push(answer);
  }
  localStorage.setItem('answers', JSON.stringify(answers));
}

function renderPreviewPage() {
  // Get questions
  var questions = getQuestions();
  $('body').append(JSON.stringify(questions));

  for (var item in questions) {
    var current = questions[item];

    if (current.type === 'yesNoQuestion') insertYesNo($('#main'), current, false);
    if (current.type === '3starRating') insert3star($('#main'), current, false);
    if (current.type === 'multipleChoiceQuestion') insertMultipleChoice($('#main'), current, false);
    if (current.type === '5starRating') insert5star($('#main'), current, false);
    if (current.type === 'rangeQuestion') insertSlider($('#main'), current, false);
    if (current.type === 'commentbox') insertcommentbox($('#main'), current, false);
  }

  $('#send-preview').click(function(){
    saveAnswers();
  });
}

```

```
});  
  
}  
  
function main() {  
  var editable = $('body').attr('data-editable') === "true";  
  
  if (!editable) { // On preview page  
    renderPreviewPage();  
  }  
  else { // On edit page  
    renderEditPage();  
  }  
  
}  
main();
```

---