AN AJAX BASED DESIGN AND IMPLEMENTATION OF A SELF
MOTIVATING SYSTEM FOR STUDENT LEARNING

A Thesis
Presented to the
Faculty of
San Diego State University

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

by
Mahesh Kumar Gudiyadolu
Spring 2012
SAN DIEGO STATE UNIVERSITY

The Undersigned Faculty Committee Approves the

Thesis of Mahesh Kumar Gudiyadolu:

An AJAX Based Design and Implementation of a Self Motivating System for

Student Learning

Subrata Bhattacharjee, Chair
Department of Mechanical Engineering

Roger Whitney
Department of Computer Science

Christopher Paolini
Department of Electrical and Computer Engineering

April 20, 2012
Approval Date
DEDICATION

I wholeheartedly dedicate this thesis to Prof. Subrata Bhattacharjee, my family and friends. Their support and wishes provided me the much needed encouragement and confidence to work hard. And last but not the least thanks to the invisible power of God for all the blessings and wonderful life given by him.
ABSTRACT OF THE THESIS

An AJAX Based Design and Implementation of a Self Motivating System for Student Learning
by
Mahesh Kumar Gudiyadolu
Master of Science in Computer Science
San Diego State University, 2012

The Expert System for Thermodynamics (TEST) is a web application that is developed to help educators create thermodynamic engineering problems and for the students to analyze and solve the problems to gain knowledge in the thermodynamics field. The TEST web application is categorized into 15 chapters.

TEST is used as effective and efficient educating medium around the world with thousands of students and educators registered with the web application. Because of the large number of students and educators accessing the TEST web application, servers are located throughout the world that makes sure a secure connection is established to cater to the user requests.

The type of registered users on TEST has been categorized into three types viz. Students, Educators and Administrators apart from the unregistered Guest users. The Guest users can just browse through the problems but cannot solve them and no history is maintained for these users.

The aim of this thesis is to provide the educators an effective interface to edit the problem attributes that makes the thermodynamic engineering problems more understandable to the students and guides the students to gain knowledge in the predefined outcomes (knowledge areas). The outcomes are categorized as follows: (1) free body diagram, (2) mass equation, (3) energy equation, (4) entropy equation, (5) ideal gas model, (6) basic cycle concept, and (7) unit conversion, etc.

In this thesis the problems has been divided into three types: Classic, Grade Me and Key problems. The educators can split the answer into parts if the answer to a particular problem has more than one answer. Successful solving of Key problems by students will guide them to solve more problems in the particular outcome to gain more knowledge. As the students solve the problems, there score is updated into the database by which the student can see his score in various outcomes on the TEST website.

The educator can also provide notes to the students for successful or wrong solutions, so that the students can take further steps based on the notes provided by the educator.

The technologies used in the development of this thesis are CSS, JavaScript, JSON, PHP and MYSQL. CSS is used to handle the styling part of the editor, JavaScript is used to dynamically create the editor, JSON format is used for problem’s attributes to be stored into the MYSQL database and PHP is used as server side language.

Finally educators, who used TEST in their engineering thermodynamics classes, will be requested to share the grade of students (anonymously). If we can compare students who

v
used SMS with those who did not and show that SMS made a difference, this will change the word as far as thermodynamics education goes.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT ...............................................................................................................................v</td>
</tr>
<tr>
<td>LIST OF TABLES ................................................................................................................... ix</td>
</tr>
<tr>
<td>LIST OF FIGURES ................................................................................................................ x</td>
</tr>
<tr>
<td>ACKNOWLEDGEMENTS ........................................................................................................ xi</td>
</tr>
<tr>
<td>CHAPTER</td>
</tr>
<tr>
<td>1 INTRODUCTION .................................................................................................................1</td>
</tr>
<tr>
<td>2 WEB APPLICATIONS ...........................................................................................................3</td>
</tr>
<tr>
<td>3 WEB TECHNOLOGIES ..........................................................................................................5</td>
</tr>
<tr>
<td>3.1 HTML ...........................................................................................................................5</td>
</tr>
<tr>
<td>3.2 AJAX ............................................................................................................................5</td>
</tr>
<tr>
<td>3.3 JavaScript ....................................................................................................................6</td>
</tr>
<tr>
<td>3.4 JSON ............................................................................................................................7</td>
</tr>
<tr>
<td>3.5 DOM .............................................................................................................................8</td>
</tr>
<tr>
<td>3.6 CSS ..............................................................................................................................9</td>
</tr>
<tr>
<td>3.6.1 CSS ID Selector ........................................................................................................9</td>
</tr>
<tr>
<td>3.6.2 CSS Class Selector ..................................................................................................9</td>
</tr>
<tr>
<td>3.7 PHP ..............................................................................................................................10</td>
</tr>
<tr>
<td>4 THE EXPERT SYSTEM FOR THERMODYNAMICS – TEST .............................................11</td>
</tr>
<tr>
<td>4.1 TEST Introduction .......................................................................................................11</td>
</tr>
<tr>
<td>4.2 Problems Module .........................................................................................................12</td>
</tr>
<tr>
<td>4.3 Design ..........................................................................................................................13</td>
</tr>
<tr>
<td>5 EDITOR ..............................................................................................................................16</td>
</tr>
<tr>
<td>5.1 Chapter View ...............................................................................................................16</td>
</tr>
<tr>
<td>5.2 Editor View ..................................................................................................................17</td>
</tr>
<tr>
<td>5.3 Student Interface ........................................................................................................24</td>
</tr>
</tbody>
</table>
6 CONCLUSION AND FUTURE ENHANCEMENTS ..................................................28
   6.1 Conclusion .......................................................................................................28
   6.2 Future Enhancements .......................................................................................28
BIBLIOGRAPHY ..............................................................................................................29
LIST OF TABLES

Table 4.1. Major Database Tables .........................................................................................15
LIST OF FIGURES

Figure 4.1. Toolbar on TEST home page. .................................................................12
Figure 4.2. Workflow diagram for forum .............................................................14
Figure 5.1. Chapter view ....................................................................................16
Figure 5.2. Key problems ..................................................................................17
Figure 5.3. Grade me problems ........................................................................18
Figure 5.4. Editor interface ................................................................................19
Figure 5.5. Problem type ...................................................................................19
Figure 5.6. Answer type .....................................................................................20
Figure 5.7. Difficulty rating ..............................................................................20
Figure 5.8. Split answers ...................................................................................21
Figure 5.9. Correct answer notes .......................................................................22
Figure 5.10. Wrong answer notes .....................................................................22
Figure 5.11. Save problem ................................................................................23
Figure 5.12. Publish problem ...........................................................................23
Figure 5.13. Take the challenge .......................................................................24
Figure 5.14. Correct answer .............................................................................25
Figure 5.15. Wrong answer ...............................................................................26
Figure 5.16. My TEST ratings hyperlink ............................................................26
Figure 5.17. My TEST ratings table .................................................................27
Figure 5.18. My TEST ratings graph .................................................................27
ACKNOWLEDGEMENTS

First and foremost, I am very grateful and my sincere thanks to the members of my thesis committee: Dr. Subrata Bhattacharjee, for taking time and guiding me through the completion of this thesis and providing invaluable encouragement, Dr. Roger Whitney, for providing me with ideas and an outlet to pursue some of my academic interests, Dr. Christopher Paolini for his constant guidance and support. Furthermore, I thank all my professors at San Diego State University as I have learned a lot from the courses they taught and all my colleagues and friends for all the helpful comments and discussions. Special thanks to my roommates who have given me a home away from home, their unconditional support and encouragement during the tough times when it was needed the most.
CHAPTER 1

INTRODUCTION

TEST, The Expert System for Thermodynamics, is a web-based portal developed at San Diego State University under the guidance of Dr. Subrata Bhattacharjee for the education of engineering thermodynamics. This web portal contains pages that depict schematics, equations, animations, thermodynamic explanations, and embedded Java Applets, called daemons, organized in a hierarchical tree structure. More and more students and educators are using TEST worldwide.

This thesis focuses on designing a Web portal for thermodynamics community where educators of the thermodynamics can edit the problem’s attributes so that the problems are effective and the students can learn and solve the thermodynamic engineering problems. The website is developed based on the AJAX technology which makes the editing of the problems without refreshing the web page. Educators can post different types of problems such as the Grade Me, Key problems in which answer is divided into parts.

Each problem will educate student in a particular outcome, as the student solves problems on the TEST site, his points are recorded into the database. The student can later view his scores that are represented based on each outcome. The score is displayed in tabular and bar chart format.

This thesis report begins with an introduction to the latest technologies that help in building the next generation web applications, their use and their massive contribution towards information sharing. Instructions are provided on how the educators can use the portal to post questions and edit them, which can be solved by students. Finally, I conclude this thesis with future enhancements that can be done on this thesis.

An Editor is web interface that is used to modify the problem attributes. The problem can be converted to a Grade Me or a Key problem in which the answer is divided into parts. Solving of a Key problem will guide the student to solve more problems in the same outcome to gain more knowledge and follow the path to become an expert in the thermodynamic engineering field.
The Educator can also provide notes for students answering the problems. In case of a correct solution, the notes may provide information to student on which problems to solve next to gain more knowledge in the same outcome. In case of a wrong solution the notes will tell student which outcomes to study to solve the problem successfully next time.

The student can also see his TEST ratings that show the problems he has solved in each of the outcomes and also his points in those outcomes. By this the student can analyze his knowledge base in the outcomes and take steps to become an expert in the Thermodynamic engineering field by solving all the problems in different outcomes. TEST ratings contain both tabular view and bar chart that shows each outcome and the points gained by student in each of the outcome.
CHAPTER 2

WEB APPLICATIONS

A web application is software accessed over the Internet from desktop computers or smart phones. One of the main reasons for web’s popularity is its ability to be updated and maintained without having the necessity of manually distributing or installing on millions of devices worldwide [1].

Web applications are written using markup language called Hypertext Mark Up Language (HTML). HTML tags are used to define the structure of the web page, Cascading Style Sheets (CSS) is used to style the web page like adding a color, images, borders, pop up windows, etc. thus making the web application look attractive, classy and elegant. While HTML and CSS are used to present information and style them on the web page, the JavaScript or ECMA Script is used to define and control the way the users can use the website and make the web application respond to user events [1]. HTML, CSS and JavaScript are defined extensively later in Chapter 3.

Some of the benefits of a website or web application are as follows [1]:

- Web applications do not require any complex “roll out” procedure to deploy it in large organizations. A compatible web browser is all that is needed.
- Browser applications typically require very little disk space on the client thus making it space effective.
- They require no upgrade procedures since all new features are implemented on the server and automatically delivered to the users.
- Web applications integrate easily into other server-side web procedures, such as email and searching.
- They are platform independent, i.e. they provide cross-platform compatibility in most cases (i.e., Windows, Mac, Linux, etc.) because they operate within a web browser window.
- With the advent of HTML5 and CSS3, programmers can create richly interactive environments natively within browsers. Some of the new features are native audio, video and animations like 2D and 3D graphics.
One of the disadvantages of web application is the security issue like SQL injection and XSS (cross site scripting). But with the improving technologies, the security issue is being considered seriously, more and more techniques are being invented to prevent web applications from hack attempts.
CHAPTER 3

WEB TECHNOLOGIES

Some of the popular web design and development technologies are HTML, CSS, AJAX and JavaScript that are used for client-side programming whereas PHP, Perl, ASP, JSP, etc. are used as server-side programming languages. For this thesis we have used HTML, CSS, AJAX, JavaScript, JSON and PHP.

3.1 HTML

Hyper Text Markup Language (HTML) is the main markup language for web pages. HTML elements are the basic building blocks of webpages.

HTML contains tags enclosed between angle brackets like <html>. The HTML tags are the basic building blocks of all the web applications. In HTML tags most commonly come in pairs like <h1> and </h1>, although some tags known as empty elements are unpaired, for example <img>. The first tag in a pair is the opening tag; the second tag is the closing tag. In between these tags web developers can add text, tags, comments, etc.

The purpose of a web browser is to read HTML documents and compose them into visible and audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page.

HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes, etc. It can embed scripts in languages such as JavaScript, which affect the behavior of HTML webpages [2].

3.2 AJAX

AJAX stands for Asynchronous JavaScript and XML. It is a group of interrelated web development methodologies used on the client side to create more interactive web applications. AJAX is used to create a web application that communicates with the server
and updates the web page on the client end without refreshing the current page. There is a very little lag time between the user click and the resultant output. For example, Google Maps is one of the major users of AJAX technology. There is a very little lag time when you scroll around the map, zoom in or out of a place and you don’t have to wait for a page to refresh or reload. One of the other major usages of AJAX is auto-complete search box, which is popularly used in YouTube, the world’s major video sharing web site, which provides the users with related suggestions when the search text is inputted in to the search box [3].

The AJAX engine itself handles any client-side activity such as data validation, which does not need to communicate with the server for a response. Only when the engine needs to communicate with the server for a response, the action takes place in the form of a JavaScript call to the AJAX engine [4]. This request is made asynchronously without interrupting user’s interaction with the application. Hence, the user will not even notice that an interaction with the server is taking place in the background, i.e. neither the web page is refreshed nor another page is loaded.

3.3 JavaScript

JavaScript is a prototype-based object-oriented scripting language that is dynamic, weakly typed and has first-class functions that are used to make the webpage more interactive. It is a multi-paradigm language, supporting object-oriented, imperative, and functional programming styles [5]. JavaScript is used to enable programmatic access to objects within the application. When used in conjunction with a web browser’s Document Object Model (DOM) [6], it can produce powerful dynamic HTML browser-based applications. JavaScript can be used to validate, modify content of the web page like forms, images, open new windows and write dynamic page content. JavaScript can be used with CSS to make DHTML (Dynamic Hyper Text Markup Language) pages. JavaScript only executes on page(s) that are on your browser window at any set time. When user stops viewing that page, any script that was running on it is immediately stopped [7].

JavaScript is an interpreted programming or scripting language supported by web browsers and other web tools. JavaScript is most often used for client-side web development. JavaScript code forms part of HTML page and can be used for things such as responding to user actions or performing data validation on the client side.
3.4 JSON

JavaScript Object Notation (JSON) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. JSON is derived from JavaScript programming language to represent simple data structures and associative arrays, called objects. Despite it being derived from JavaScript, it is language-independent and is available for most programming languages. JSON is built on two structures:

- A collection of name/value pairs. In various languages, this is realized as an object, record, struct, dictionary, hash table, keyed list, or associative array.
- An ordered list of values. In most languages, this is realized as an array, vector, list, or sequence [8].

The basic JSON types are:

- Object (a collection of key-value pairs, comma-separated and enclosed in curly brackets)
- Array (an ordered sequence of values, comma-separated and enclosed in square brackets)
- String (double-quoted Unicode with backslash escaping)
- Number (integer, real, or floating point)
- Boolean (true and false) [9].

JSON format is an unordered set of name-value pair object, which begins with a left brace and ends with right brace. A colon separates name and value, and a comma separates the pairs. Following is an example of JSON string that represents an object Answer used in this thesis. The object has string fields for value, units, minimum, maximum and points.

```json
{
  "answers": [
    {
      "val": "1.33 ",
      "units": "kJ",
      "min": "1.2",
      "max": "1.46",
      "pts": "10"
    },
    {
      "val": "1333 ",
      "units": "kPa",
      "min": "1199.7",
      "max": "1466.3",
      "pts": "10"
    }
  ]
}
```
Suppose the above JSON string is the value of a JavaScript string variable JSON_Answer, it can be converted to object using JavaScript built in method eval() as shown below:

```javascript
Var JSONObj = eval("\" + JSON_Answer + \"\");
```

Once the JSON object has been created, the attributes or fields contained in the object can be accessed using object notation like JSONObj.val, JSONObj.units... eval() should be used only to parse JSON strings from trusted source. Parsing a wrongly formatted JSON string will produce incorrect results.

### 3.5 DOM

The Document Object Model (DOM) is a platform and language independent interface that allows programs and scripts to dynamically build documents, navigate through their structure, and add, modify or delete elements and content of documents. Programmers can access, update or delete the style and structure of any document. The document can be further processed and the results of that processing can be incorporated back into the presented page. Here, document is modeled as object that contains behavior of the document and objects inside it [10].

A W3C standard that defines a standard way to access and edit HTML documents is called HTML DOM. It provides an object model and programmatic interface for HTML. It also defines objects, properties and functions for HTML elements. The entire HTML document can be presented in a tree structure. Everything in a HTML document is considered as a node. All these nodes have a hierarchical relationship to each other. The nodes can be accessed with any programming language. Any node in HTML DOM can be accessed either by its ID or tag name or navigating the entire node tree [10]. Document is root object and we can use different methods defined in the DOM to access any node by
simply passing the ID or tag name to the calling function. Once we get node reference, we can use different methods of node object to modify HTML DOM at runtime by programming languages like JavaScript and provide user with better client side experience.

3.6 CSS

Cascading Style Sheets (CSS) were designed for separating the content from its presentation. Once the browsers started supporting many tags, it became necessary to create web sites where the content of HTML documents was separated from the presentation. User can define colors, fonts, layout, and other aspects of document presentation in the CSS [11].

Every HTML element can be assigned a specific set of styles, which are defined in the CSS and apply it to as many web pages as you want. All elements in the web page having the same style can be updated automatically by only making one change in the style sheet. Style can be either attached as a separate document or embedded in the HTML document. Also, multiple style sheets can be imported in the same HTML document. CSS Styles can be defined in various ways [12]:

1. Inline styles, i.e. next to the element itself.
2. Internal styles, i.e. inside the <head> tag
3. External styles, i.e. in a separate file.

3.6.1 CSS ID Selector

The ID selector is used when we have to specify a style for a single and unique element.

The ID selector uses the ID attribute of the HTML element, and is defined with a “#”. The style rule below will be applied to the element with id="div1":

```
#div1 {text-align: center ;}
```

3.6.2 CSS Class Selector

The class selector is when we have to specify a style for more than one element. This allows you to set a particular style for many HTML elements with the same class.

The class selector uses the HTML class attribute and is defined with a “.” (dot).

In the example below, all HTML elements with class="center" will be center-aligned:

```
.center {text-align: center;}
```
3.7 PHP

PHP is a general-purpose server-side scripting language originally designed for Web development to produce dynamic Web pages. It is one of the first developed server-side scripting languages to be embedded into an HTML source document, rather than calling an external file to process data. Ultimately, the code is interpreted by a Web server with a PHP processor module which generates the resulting Web page. It also has evolved to include a command-line interface capability and can be used in standalone graphical applications. PHP can be deployed on most Web servers and also as a standalone shell on almost every operating system and platform free of charge. A competitor to Microsoft’s Active Server Pages (ASP) server-side script engine and similar languages, PHP is installed on more than 20 million Web sites and 1 million Web servers [13].

Rasmus Lerdorf originally created PHP in 1995. The main implementation of PHP is now produced by The PHP Group and serves as the formal reference to the PHP language. PHP is free software released under the PHP License, which is incompatible with the GNU General Public License (GPL) due to restrictions on the usage of the term PHP [13].

While PHP originally stood for “Personal Home Page”, it is now said to stand for “PHP: Hypertext Preprocessor”, a recursive acronym [13].
CHAPTER 4

THE EXPERT SYSTEM FOR THERMODYNAMICS – TEST

4.1 TEST INTRODUCTION

TEST, The Expert System for Thermodynamics, is a web-based tool developed at San Diego State University for spreading the knowledge about engineering thermodynamics. TEST consists of thousands of HTML pages combined with the power of Java and AJAX programming to offer a visual environment for students and educators to analyze and understand thermodynamic problems. These pages have schematics, equations, animations, thermodynamic explanations, and embedded Java Applets, called daemons, organized in a hierarchical tree structure that makes TEST a general-purpose visual tool for solving thermodynamic problems and performing what-if scenarios online for students and professionals [14].

TEST is an extremely powerful, unique and effective web ware for learning, teaching, and practicing thermodynamics. TEST consists of 16 chapters which covers the entire range of topics taught in most engineering thermodynamic courses. TEST stands out of the other available software’s owing to its visual effects which allow users to visualize thermodynamic principles and systems through animations and solve complex problems using its daemons and RIAs. EES [15] is a comprehensive thermal-science programming language that runs in the Microsoft Windows environment. TPX [16] is an Excel plug-in where the core thermodynamic state can be evaluated by entering two independent properties. Thermoptim [17] is Java application for analyzing thermodynamic cycles and can be run over the web. The Qualitative Reasoning Group at Northwestern University has developed a Windows installable application named CyclePad [18], which allows users to construct and analyze a wide variety of thermodynamic cycles. None of these packages, however, are comprehensive enough to cover the diverse range of topics covered in engineering thermodynamics.
4.2 Problems Module

TEST is divided into different modules for ease of access to its users, i.e. educators and students. Problems are one of the modules and require a user login to access the Forum. Problems tab is present in the navigation bar present at the bottom of the web page as depicted by Figure 4.1.

![Figure 4.1. Toolbar on TEST home page.](image)

The tool bar in Figure 4.1 makes moving to another module via just one mouse click. This bar is always stationed at the bottom of page no matter what module you are accessing, hence making every part of website quickly accessible for user.

The Problems section has a layout of a typical questions and answers section. Given the variety of thermodynamics topics, Problems has been organized based on the topics, i.e. it has been divided into sub sections. The problems section also shows a diagram to the student if the educator has uploaded it.

The Problems module has been designed in order to provide a very user-friendly experience with more user controls. Students can select a problem by clicking on the problem. The educator can edit the problem by clicking on the Edit Problem hyperlink, which is visible only if the user logged in is identified as Educator. When educator clicks on edit problem hyperlink, the entire problem attributes like Id, Title, Answer, ProblemType, AnswerType, Image, etc. will be fetched from database via AJAX and shown in the nicely designed editor and again without refreshing the web page as AJAX is used. Now the educator can change the problem attributes and save it to the database and publish it live for the students to take up the challenge and solve the problem.

Following are the features available for the Educators when using this Problems section:

1. Edit the problem parameters such as Problem Type, Answer Type, Title, Answer and Hint, etc.
2. Save the problem to the database.
3. Publish the problem to live.
4. Split the answer into parts.
5. Add the keywords describing the problem.
6. Provide notes or suggestions for students solving the problems.

   The problem attributes that can be edited by the educator as follows:

1. Answer Type
2. Problem Type
3. Difficulty Rating
4. Keywords
5. Outcomes
6. Special Title
7. Question
8. What-if
9. Hint
10. Answers
11. Correct Answer Notes
12. Wrong Answer Notes
13. Answer Hidden to.

### 4.3 Design

The first step of the project was to build a web service to facilitate the data transfer between the browser and the database. This synchronizer is called from the front end to make a web service call. The parameters are sent to the backend PHP file which builds a query dynamically based on the request from the client and queries the database and then sends back the results. These results are rendered on the front end to the user by chapter.js file by building the page with HTML tags dynamically. See Figure 4.2 for workflow diagram for forum.

Major database tables involved in the project are: problem, section, outcomes and userstats. Table 1 shows the structure of each of these tables.

The problem table contains all the attributes of the problems. Every time a chapter is accessed, the problems and its attributes are fetched from this table.

The section table contains the section related information, i.e. the section id, section title and all the problems ids that belong to each of the section.
The outcomes table contains the title, description and problem ids that belong to each of the outcome.

The userstats table contains the user related information like email, number of logins and also contains the field jSolvedProblems, which is used to store all the problem ids that the user has solved.
Table 4.1. Major Database Tables

<table>
<thead>
<tr>
<th>Database Table</th>
<th>Database Table</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Problem</strong></td>
<td><strong>Section</strong></td>
</tr>
<tr>
<td>pid : int</td>
<td>sid : int</td>
</tr>
<tr>
<td>pKey : string</td>
<td>sKey : string</td>
</tr>
<tr>
<td>pOldKey : string</td>
<td>cNumber : int</td>
</tr>
<tr>
<td>sAnswerType : string</td>
<td>sNumber : int</td>
</tr>
<tr>
<td>sProblemType : string</td>
<td>pids : string</td>
</tr>
<tr>
<td>sKeywords : string</td>
<td>sTitle : string</td>
</tr>
<tr>
<td>sOutcomes : string</td>
<td></td>
</tr>
<tr>
<td>sSpTitle : string</td>
<td></td>
</tr>
<tr>
<td>sMainBody : string</td>
<td></td>
</tr>
<tr>
<td>sWhatif : string</td>
<td></td>
</tr>
<tr>
<td>sAnswer : string</td>
<td></td>
</tr>
<tr>
<td>sAnswersHiddenTo : string</td>
<td></td>
</tr>
<tr>
<td>jSplitAnswers : json string</td>
<td></td>
</tr>
<tr>
<td>sHint : string</td>
<td></td>
</tr>
<tr>
<td>sCorrectAnswerNotes : string</td>
<td></td>
</tr>
<tr>
<td>sWrongAnswerNotes : string</td>
<td></td>
</tr>
<tr>
<td>iLinkType : int</td>
<td></td>
</tr>
<tr>
<td>iLinkSrcType : int</td>
<td></td>
</tr>
<tr>
<td>sManUrl : string</td>
<td></td>
</tr>
<tr>
<td>sTestURL : string</td>
<td></td>
</tr>
<tr>
<td>sLinkURL : string</td>
<td></td>
</tr>
<tr>
<td>sLinkSrcURL : string</td>
<td></td>
</tr>
<tr>
<td>bImgLarge : string</td>
<td></td>
</tr>
<tr>
<td>bTestIcon : string</td>
<td></td>
</tr>
<tr>
<td>iRating : string</td>
<td></td>
</tr>
<tr>
<td>sWhatif : string</td>
<td></td>
</tr>
<tr>
<td>sOutcome : string</td>
<td></td>
</tr>
<tr>
<td>pids : string</td>
<td></td>
</tr>
<tr>
<td>sDescription : string</td>
<td></td>
</tr>
<tr>
<td>sSpTitle : string</td>
<td></td>
</tr>
<tr>
<td>id : int</td>
<td></td>
</tr>
<tr>
<td>sMainBody : string</td>
<td></td>
</tr>
<tr>
<td>sWhatif : string</td>
<td></td>
</tr>
<tr>
<td>sAnswer : string</td>
<td></td>
</tr>
<tr>
<td>sAnswersHiddenTo : string</td>
<td></td>
</tr>
<tr>
<td>jSplitAnswers : json string</td>
<td></td>
</tr>
<tr>
<td>sManUrl : string</td>
<td></td>
</tr>
<tr>
<td>sTestURL : string</td>
<td></td>
</tr>
<tr>
<td>solutionman : string</td>
<td></td>
</tr>
<tr>
<td>sLinkURL : string</td>
<td></td>
</tr>
<tr>
<td>sLinkSrcURL : string</td>
<td></td>
</tr>
<tr>
<td>bImgLarge : string</td>
<td></td>
</tr>
<tr>
<td>bTestIcon : string</td>
<td></td>
</tr>
<tr>
<td>iRating : string</td>
<td></td>
</tr>
<tr>
<td>email : string</td>
<td></td>
</tr>
<tr>
<td>comments : string</td>
<td></td>
</tr>
<tr>
<td>no_of_logins: string</td>
<td></td>
</tr>
<tr>
<td>last_login : timestamp</td>
<td></td>
</tr>
<tr>
<td>subscription: string</td>
<td></td>
</tr>
<tr>
<td>book : string</td>
<td></td>
</tr>
<tr>
<td>blogprofile : string</td>
<td></td>
</tr>
<tr>
<td>userprofile : string</td>
<td></td>
</tr>
<tr>
<td>jSolvedProblems : string</td>
<td></td>
</tr>
</tbody>
</table>
CHAPTER 5

EDITOR

This chapter explains the layout of the Editor and various features available for the educators and students.

5.1 CHAPTER VIEW

When a particular chapter is accessed, all the problems belong to this chapter is fetched from the database and displayed nicely in the browser. The chapter is divided into sections where each section presents problems that will educate the student in a particular thermodynamic engineering field.

The following snapshots will depict the chapter view.

Figure 5.1 shows the problems for Chapter 2, each problem is fetched from the database based on the chapter number.

Figure 5.1. Chapter view.
Figure 5.2 shows how a key problem is displayed on the TEST site. The key icon makes it easy for students and educators to differentiate the different problems. Hovering over the icon shows a description about the key problems.

Figure 5.2. Key problems.

Figure 5.3 shows how a Grade Me problem is displayed on the TEST site. The pencil icon makes it easy for students and educators to differentiate the different problems. Hovering over the icon shows a small description about Grade Me problems.

5.2 EDITOR VIEW

Each problem in the chapter will have an Edit Problem hyperlink, when clicked on it, a nicely designed editor is displayed which allows the educator to edit the problem attributes, convert the problem to a key problem, split the answer into parts, publish the problem to live for students to solve it, etc.

The following snapshots will depict the Editor view and tasks that can be performed by the educator using the editor.
Figure 5.3. Grade me problems.

Figure 5.4 shows the editor for the educator to edit the problem’s attributes. The tasks that can be done by using the editor interface are edit the attributes of a problem, make problem a key or grade me problem, add outcomes that define which areas of knowledge the problem covers, assign a difficulty rating to the problem, etc.

Figure 5.5 shows how a problem’s type can be changed. The educator can choose one of the three problem types like Classic, Key or Grade Me from the dropdown as shown in the Figure 5.5.

By clicking on Split Answer button, the problem’s answer can be split into parts as shown in Figure 5.6. This will save time for the educator as it automatically split the answer into parts.

Figure 5.7 shows the educator can assign a difficulty rating between 1 and 5 for each problem by choosing from the dropdown. The difficulty rating can be used in the future enhancements to suggest a problem for the student to solve.
Figure 5.4. Editor interface.

Figure 5.5. Problem type.
Figure 5.6. Answer type.

Figure 5.7. Difficulty rating.
When the educator clicks the Split Answer button for a problem, the table as shown in Figure 5.8 is generated automatically by splitting the answer into parts. The table depicts the value, unit and points for each answer part.

Figure 5.8. Split answers.

Figure 5.9 shows the educator can also provide notes to the student if the problem has been solved successfully, e.g. the notes may instruct the student which problems to solve next to gain more expertise in the same outcome of thermodynamic field.

Figure 5.10 shows the educator can also provide notes to the student if a particular problem has been answered incorrectly, e.g. the notes may instruct the student which chapters to study to be able to solve the problems correctly next time.

The educator can save the problem to the database table; the save will update each of the attributes to the table. The problem saved is not visible to the student until it is published to live by the educator which is depicted in the next Figure 5.11.

Figure 5.12 shows the educator can publish the problem to the live database table; publish will take the problem from development table, save to the live table, and the student would be able to see the problem and, if desired, can take up the challenge of solving it.
Figure 5.9. Correct answer notes.

Figure 5.10. Wrong answer notes.
Figure 5.11. Save problem.

Figure 5.12. Publish problem.
5.3 STUDENT INTERFACE

As the students solve problems, the problem ID the student has solved, its points, the number of attempts the students took is recorded in the database. Later when the students want to see how they have fared in the thermodynamics problem solving on the TEST site, he can view his TEST ratings by clicking on the hyperlink My TEST Ratings, which will show the students points in each of 24 outcomes, the total number of problems in each outcome and the number of problems the student has solved in each outcome.

The following snapshots will depict the student interface of the TEST site.

When the student wants to solve a key problem and clicks on the key icon, Figure 5.13 interface is presented to the student where student can enter the answers for the problem. After the student has finished entering the answers, student can click on the Grade it button to validate the answers.

![Figure 5.13. Take the challenge.](image)

Each of the answer entered by the student is validated with the answers present in the database table which is identified by the problem id; Figure 5.14 shows that a student has solved the problem correctly.
Figure 5.14. Correct answer.

Figure 5.15 show that the student has answered a problem wrongly. If desired the student can try again. The number of attempts taken by the student to solve the problem is recorded in the database table.

Figure 5.16 show the hyperlink My TEST Ratings, when clicked on, the student’s problem solving history, i.e. his score in each of the outcomes, total problems in each outcome and the number of problems solved by student in each outcome is shown to the student in a tabular format.

Figure 5.17 show the tabular format of the students score and outcomes. The table has fields like Points, Total Problems and Solved Problems for each of the outcomes.

Figure 5.18 show the tabular and graph format of the student score and outcomes. The table has fields like Points, Total Problems and Solved Problems for each of the outcomes.

The graph is plotted with the X axis being the outcome ID and the Y axis being the total points the student has scored in the outcome. Figure 5.18 show that the student has 5 points each in each of outcomes 1, 6 and 12.
Figure 5.15. Wrong answer.

Figure 5.16. My TEST ratings hyperlink.
Figure 5.17. My TEST ratings table.

Figure 5.18. My TEST ratings graph.
CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENTS

6.1 CONCLUSION

This thesis sets an initiative towards creating an effective way for educators to post problems to be solved by students to gain knowledge in the Thermodynamics field. A simple design and easy to understand features allows the instructors and students do their tasks with ease. The entire website is built using open source technologies thus making the website accessible from any device that has Internet connection and thereby making it cost effective as well. No additional software or hardware upgrades are needed for this web application to work.

6.2 FUTURE ENHANCEMENTS

After the development of Web Forum, I feel there is a scope for following enhancements in the future:

- To allow the student to search for a problem based on chapter or section or outcome name or date posted, etc.

- Currently when the problems are suggested for the student to solve, the problem ids are presented to him, in the future those problem ids can be made hyperlinks which will take the user to the chapter which contains that problem that the student wants to solve.

- If the student has solved a problem in the past and the educator posts more problems that cover the same outcome, then an email can be sent to the students to notify that more problems have been posted in that particular outcome in which he may be interested.

- Send periodic email alerts to students asking them to solve problems in a particular outcome based on their problem solving history to gain more knowledge in the thermodynamic engineering field.

- Allow the students to control the email alerts.

- To allow the students to get their score report via email.
BIBLIOGRAPHY


