

View Component of Web-based IDE in CakePHP

A Writing Project

Presented to

The Faculty of the Department of Computer Science

San José State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

By

Swathi Vegesna

Spring 2010

Abstract

The aim of the project is to build the view component of a Web-based IDE that enables users to create XHTML pages and link up the tags with the Controller files by only using drag and drop mechanism in cakePHP. This is the view component of the MVC based Web-based IDE, which will connect the Controllers and View components. This project will enhance these elements by making them dynamic which is achieved by connecting to the Controller components. This IDE helps the users to create the dynamic web pages even without good knowledge of XHTML and server side scripting languages. The user will be able to include all kinds of elements only with drag and drop mechanism and develop the whole web page.

Table of Contents

1. Introduction	4
2. Experiments with cakePHP	5
2.1 Motivation	5
2.2 Goal	5
2.3 Implementation	5
2.4 Remarks	7
3. Performance test on JavaScript frameworks	7
3.1 Motivation	7
3.2 Goal	8
3.3 Implementation	8
3.4 Remarks	10
4. Drag and Drops in jQuery	10
4.1 Motivation	10
4.2 Goal	10
4.3 Implementation	10
4.4 Remarks	12
5. Basic Layout of the IDE	13
5.1 Motivation	13
5.2 Goal	13
5.3 Implementation	13
5.4 Remarks	16
6. Conclusion.....	16
References.....	17

1. Introduction

An Integrated development environment (IDE) is a software application that provides comprehensive facilities to computer programmers for software development. Some of the features of an IDE include, a source code editor, a compiler and/or an interpreter, build automation tools and a debugger. There are two categories of IDE's; one is the traditional Desktop-based IDE and the other is the newly emerging Web-based IDE.

A Web-based IDE is a browser-based code development environment which can be accessed with a connection to the server and the available browser. This does not need any installation on the local developer workstation which in terms reduces hardware costs and management overhead while ensuring everyone with the latest version and patches. Consequently, developers can make code changes from anywhere, using any machine.

The proposed project is to develop the GUI (Graphical User Interface) for a web-based IDE connecting the Controller and View components. This IDE is used for creating dynamic XHTML pages which is achieved by connecting the static XHTML elements with the controller functions. The developer should be able to develop a complete web page only by clicking, dragging, and dropping of the elements on the IDE. One such web-based IDE is the "Drag and drop site creator" by SQLFusion, LLC. In this, the developer can create all the XHTML elements only by dragging and dropping the components in the GUI of the IDE ^[2]. One main drawback of the Drag and drop site creator is that the user can only generate the static pages. This problem is solved by the proposed IDE as the developer is given an option to connect view and controller components.

The technologies which will be used in building the project include HTML, CSS and JavaScript as the front end, PHP as the backend script and MySQL as the database. In order to build the project in the MVC (Model View Controller) architecture I have chosen cakePHP as my framework. As a part of CS297 I have learned the above stated technologies and developed some projects in demonstrating the understating of the concepts. My work is categorized into four deliverables; namely, Experiments with CakePHP, Performance on JavaScript frameworks,

Drag and Drops in jQuery and Basic layout of IDE. This report will describe the four deliverables including their goals and implementation, conclusion and future work.

2. Experiments with cakePHP

2.1 Motivation

The main objective of this deliverable was to experiment with PHP and CakePHP. CakePHP is a rapid development framework for PHP and provides an extensible architecture for developing, maintaining, and deploying Web applications. CakePHP uses a MVC platform within the convention over configuration paradigm. This helps the developers write less code ^[1]. As the complete IDE needs to build on this framework, this deliverable scope was extended to create a web application.

2.2 Goal

The goal of developing the Book Collection website was to get to know the database operations and handle the data from different models. This website is where users can register to the website and login into their accounts to view the book collection available and add books to their cart for reservation.

2.3 Implementation

The website consists of the Home page, Registration page, Login page, Book collection page and Cart Page. The software needed to develop the application is WAMP (Windows Apache HTTP Server PHP MySQL Database) and CakePHP. The Home page of the website is the one where new users can enter into the Registration page and the returning users can go to the Login page.

The Registration page contains a form where a user has to enter the user id, password and first name to register onto the website. Once he clicks on the register button, the user is taken to the Login page.

The Login page is the place where the user is authenticated to view the book collection. It contains a form where the user is asked to enter his username and password. Once the user clicks on the login button entering the correct credentials he is taken to the Book collection page. The book collection page shown in Fig.1, is the place where users can view the entire collection of books. The user can add any number of books to his cart, requesting for the reservation of the book. The right user can also add a book to the collection by clicking on the *Add book* link.

Similarly, the same user can also edit the book details in the collection by clicking on the *Edit* link.

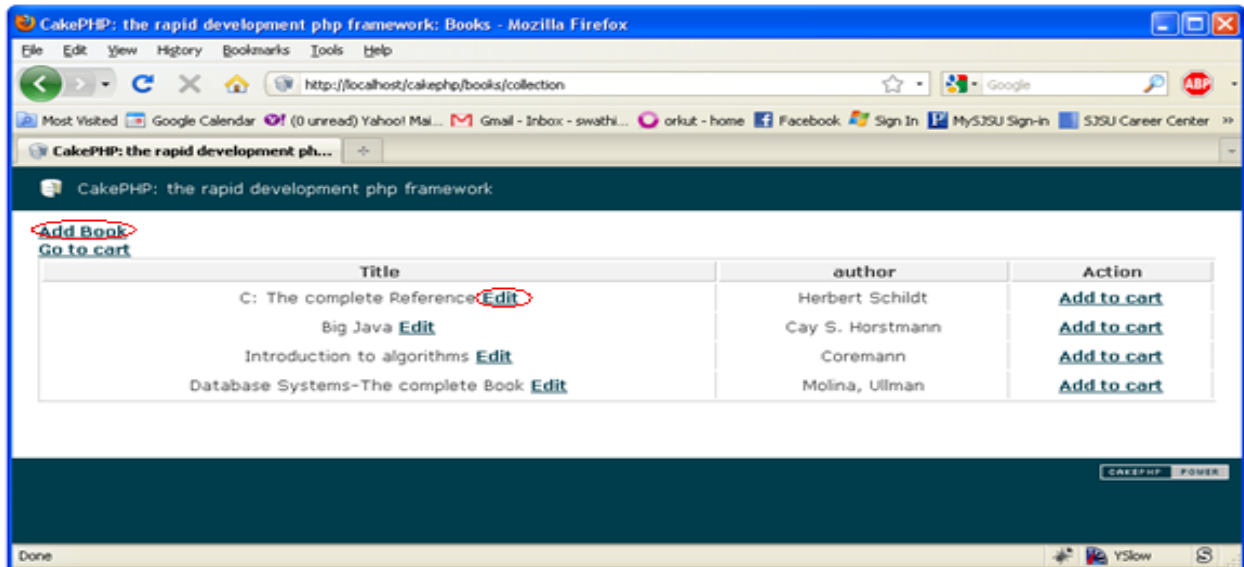


Fig. 1 Book collection page

All the users can add the books to their carts by clicking on the link *Add to cart* and also view the book in the carts by clicking the link *Go to cart*. The user can then finally logout from the website by clicking on the link *Logout*.

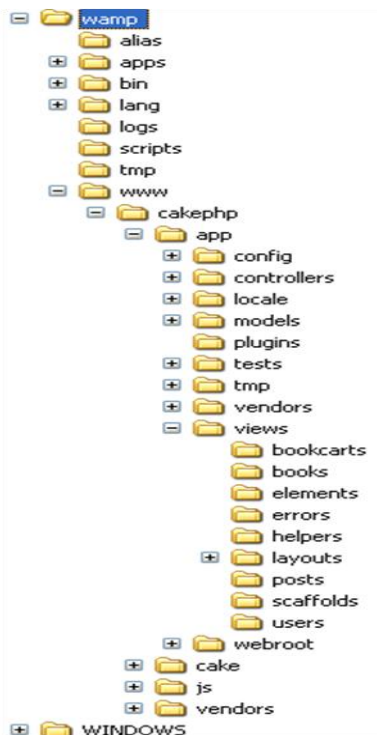


Fig. 2 Directory Tree

To build this web application the following tasks are accomplished.

1. Installed WAMP 2.0i, which includes Apache2.2.11, MySQL5.1.36 and PHP5.3.0
2. Downloaded CakePHP 1.3.0
3. Added cakephp folder to the WWW folder of WAMP.
4. Configured the database settings in the /app/config/database.php file.
5. Created three tables in the database namely users, books and book_carts (according to the naming conventions to be followed in CakePHP).(Fig.2 shows the Directory tree)
6. Created the associated model.php files in cakephp\app\models\
7. Created the associated controller.php files in cakephp\app\controllers\
8. Created the .ctp files in the associated view folders under cakephp\app\views\

2.4 Remarks

The website developed helped in gaining experience on how to work in the MVC framework and I found CakePHP to be a well organized framework wherein each file is put into its corresponding folder like models, views and controllers. This makes both the developing and maintaining the web applications easy. Also I gained knowledge in how to handle the model objects in the controllers and build the view components.

3. Performance test on JavaScript frameworks

3.1 Motivation

The main objective of this deliverable was to conduct different performance tests on the various JavaScript frameworks to decide on the framework to be used to build the GUI of the web-based IDE. To build the most effective GUI we need to pick up the framework with the best features. Some of the best features are fast, compact with a concise library, easy for building dynamic web applications, less code writing, uses less memory and is compatible with all browsers.

Some of the most famous JavaScript frameworks include

jQuery - This is a fast, concise, JavaScript Library that simplifies how you traverse HTML documents, handle events, perform animations, and add Ajax interactions to your web pages. jQuery is designed to change the way that you write JavaScript.

MooTools - This is a compact, modular, Object-Oriented JavaScript framework designed for the intermediate to advanced JavaScript developer. It allows you to write powerful, flexible, and cross-browser code with its elegant, well documented, and coherent API.

Prototype - This is a JavaScript Framework that aims to ease development of dynamic web applications. Featuring a unique, easy-to-use toolkit for class-driven development and the nicest Ajax library around, Prototype is turning into the code-base of choice for web application developers everywhere.

Dojo Toolkit - This is an Open Source DHTML toolkit written in JavaScript. Dojo aims to solve some long-standing historical problems with DHTML which prevented mass adoption of dynamic web application development.

script.aculo.us - This provides user with easy-to-use, cross-browser user interface JavaScript libraries to make your web sites and web applications on fly.

YUI Library - This is the most robust framework which is proven, consistent, reliable which also contains the testing framework. Even with the heavy libraries, it only provides a very few utility or helper functions. ^[3]

To get to know how to develop web pages using different frameworks JavaScript frameworks the deliverable scope is extended to create a simple web pages in the selected four frameworks namely, jQuery, YUI, DOJO and Prototype.

3.2 Goal

The goal of conducting the performance test on the four frameworks mentioned above is to get a flavor of using them. This was achieved by building simple web pages which have functionalities like animations, drags and drops. So I built the same HTML page in all the four frameworks and conducted the speed test, weight test and memory test.

3.3 Implementation

The HTML page that was built to conduct the speed test and the weight test was an animated post card and to conduct the memory test is a HTML page with simple demonstration of drag and drop. The libraries needed to develop the above pages include the JavaScript files of jQuery, YUI library, Prototype and zip file of DOJO tool kit. To conduct the performance tests YSlow plug-in is needed and also Slickspeed tool.

The Fig. 3 shows the animated post card in jQuery taken from ^[4] and developed the same in YUI, prototype and DOJO. And conducted the speed test and weight test using the YSlow Plug-in and the results are show in the Fig. 4.

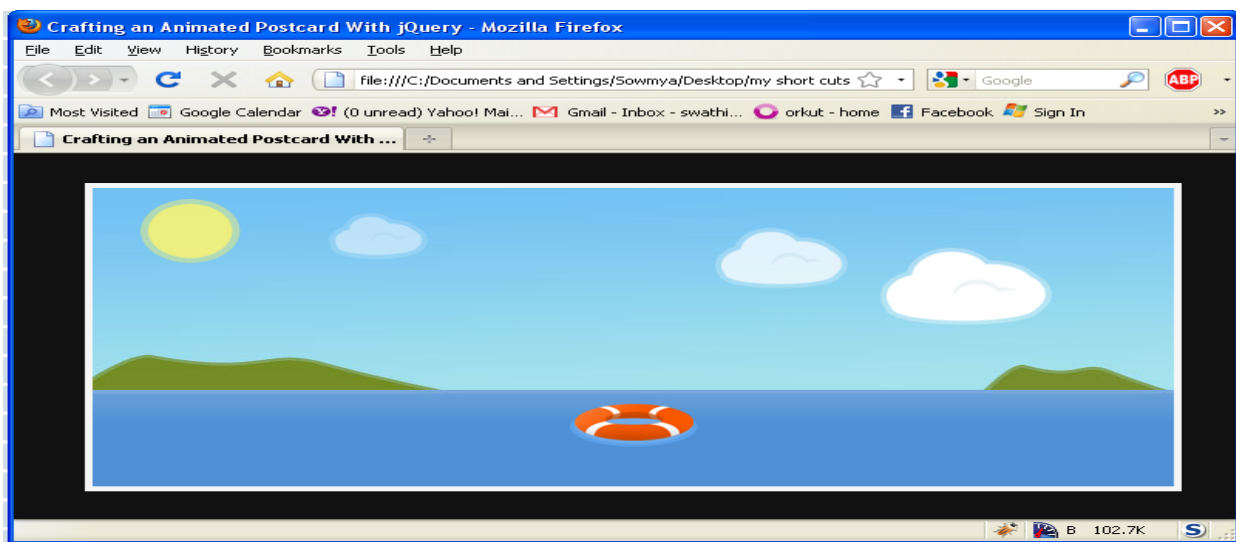


Fig. 3 Animated post card

According to the speed and weight test results of the frameworks jQuery proves to be the best one and YUI and prototype are the ones with poor performance.

JavaScript frameworks	Grade	Performance score	No. of HTTP requests	Weight
jQuery	A	84	10	10.2K
YUI	B	76	14	97.2 K
DOJO	A	80	11	76.4 K
Prototype	B	78	12	40.5 K

Fig. 4 Speed and Weight Test results

Slickspeed tool conducts another performance test where every framework runs in its own iFrame, to avoid conflicts. Tests are run selector by selector, with an interval to prevent the browser from freezing. Tests are run in a neutral environment, with no inclusion of any library or framework [4]. Fig. 5 shows the test results.

selectors	MooTools 1.2	jQuery 1.2.6	Prototype 1.6.0.2	YUI 2.5.2 Selector beta	Dojo 1.1.1
body	1 ms 1 found	1 ms 1 found	12 ms 1 found	1 ms 1 found	2 ms 1 found
div	2 ms 51 found	1 ms 51 found	9 ms 51 found	1 ms 51 found	2 ms 51 found
body div	2 ms 51 found	2 ms 51 found	18 ms 51 found	2 ms 51 found	2 ms 51 found
div p	4 ms 140 found	5 ms 140 found	13 ms 140 found	11 ms 140 found	2 ms 140 found
div > p	3 ms 134 found	5 ms 134 found	11 ms 134 found	4 ms 134 found	1 ms 134 found
div + p	2 ms 22 found	3 ms 22 found	12 ms 22 found	8 ms 22 found	2 ms 22 found
div ~ p	10 ms 183 found	11 ms 183 found	43 ms 183 found	8 ms 183 found	3 ms 183 found
div[class^=exa][class\$=mple]	3 ms 43 found	2 ms 43 found	10 ms 43 found	2 ms 43 found	2 ms 43 found
div p a	4 ms 12 found	5 ms 12 found	13 ms 12 found	19 ms 12 found	2 ms 12 found
div, p, a	9 ms 671 found	11 ms 671 found	33 ms 671 found	13 ms 671 found	2 ms 671 found
.note	16 ms 14 found	15 ms 14 found	8 ms 14 found	36 ms 14 found	1 ms 14 found
div.example	2 ms 43 found	1 ms 43 found	10 ms 43 found	2 ms 43 found	2 ms 43 found

Fig. 5 Slickspeed results

Even in this test Prototype and YUI are the ones with poor performance and jQuery and DOJO are the ones with the best performance.

Memory test was conducted on the pages with simple drag and drops with the help of the Windows Task Manager. The memory usage of Firefox is noted down by opening each page in several tabs. Even in this test jQuery was successful enough to be the best framework.

When it comes to Compatibility issues with different browsers DOJO is not supported in the IE 8, and also has a poor performance in Safari and Opera.

3.4 Remarks

The results that were obtained after conducting various performance tests made me decide to choose jQuery as my JavaScript framework for the IDE. Moreover it is the mostly widely used framework, has powerful and short code and is highly compatible with all the browsers. This deliverable also helped me get a taste of all the famous frameworks and also in choosing the right framework for the project.

4. Drag and Drops in jQuery

4.1 Motivation

The main objective of this deliverable was to experiment with Drag and Drop functionalities in jQuery. The proposed IDE should be so developed that the developers can create the dynamic XHTML pages only using some clicks and drag and drops. As these drag and drops are vital for the project, getting hands-on experience with these extends the scope of the deliverable to build a GUI which includes the implementation of all the functionalities provided by the Draggable and Droppable classes in jQuery.

4.2 Goal

The goal of this deliverable was to develop a GUI similar to that of Microsoft Access and learn how to obtain drag and drops using jQuery. The GUI consists of two main components; one is to list the database structure with all the tables and the other is working area where the new tables can be created by connecting them with listed tables. The relation between the tables is shown by a line that is pointing to the new table. The user should also be able to add or delete databases and tables.

4.3 Implementation

There are three main functionalities that have been developed into the GUI; Namely, Developing the Tree-view of the tables, Creating the Draggable and Droppable objects and Representing the relation by drawing a line. The open source libraries required to develop the above functionalities are jquery.min.js, jquery.treeview.js, jquery-ui.js and wz_jsgraphics.js.

Developing the Tree – View of the tables: In order to develop the GUI shown in the Fig.6, the body of the HTML page should contain the unordered list of table names. Secondly on page load

we need to call the function `treeview()` on the unordered list. The following is the script to be included in the HTML page.

```
$(document).ready(function(){
    $("#tablelist").treeview();
});
```



Fig. 6 Tree-view of the tables

Creating the Draggable and Droppable objects: In order to create a Draggable object we need to call `draggable()`, to add extra features to it we can use some options like `helper`, `cursor`, `drag` etc, and specify the action needed. In this case in order to get a clone of the object that is dragged and the change of cursor I used `clone` and `move` as my values to the options.

```
$(".table").draggable({
    helper:'clone',
    cursor: 'move'
});
```

In order to create a Droppable object we need to call `droppable()`, to specify what elements it should accept we can specify by using the `accept` option. The action code of what should happen after the specified object is been dropped should be written In the `drop` option. The following code will create a droppable object on element with id `area` and once a table is dropped on it, `displayTable(id)` function is invoked.

```
$("#area").droppable({
    accept: '.file',
    drop: function(event, ui) {
        var d = ui.draggable.attr("id");
        displayTable(d);
    }
});
```

Representing the relation by drawing a line: In order represent the table relation by a line drawn from the source table to destination table, we need to add a drop option to the field that is been dropped to a table and invoke the draw (id, id) function.

```

$( '#'+d ).droppable({
    accept: '.f',
    drop: function( event, ui ) {
        e = ui.draggable.attr( "id" );
        draw( e, d );
    }
});

```

The Fig. 7 shows the complete working model of the GUI where in user can expand and contract the tree-view of the tables, add new databases and tables, drop the table on to view area to view the table and build some relations with new table created.

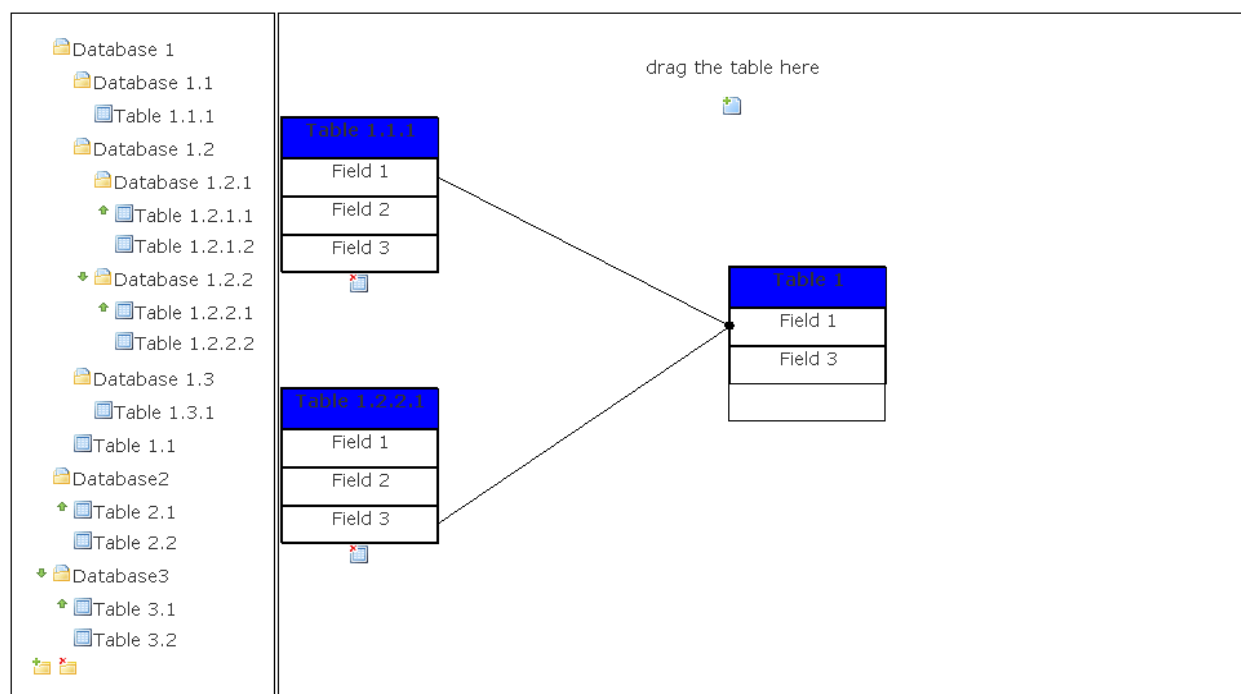


Fig. 7 GUI with Drag and Drop

4.4 Remarks

The GUI developed could successful achieve all the requirements to be similar to the Query builder in Microsoft Access. This experience in the drag and drops gave me complete

knowledge about the Draggable and Droppable classes in jQuery and the excellent features they offer.

5. Basic Layout of the IDE

5.1 Motivation

The main objective of this deliverable is to build the basic GUI for the proposed Web-based IDE with proper layout and basic drag and drop functionalities. The same layout will be used for the future work to include more functionality to all the elements in the GUI.

5.2 Goal

The goal of developing the basic layout of the IDE is to start the real working for the web-based IDE and also to implement all the basic features using the knowledge gained from the previous two deliverables.

5.3 Implementation

The basic layout developed is accomplished in three steps; Namely, Developing the basic layout, Developing the drop downs and Creating the XHTML elements.

The open source libraries required to develop the above functionalities are jquery.min.js, jquery.treeview.js, and jquery-ui.js.

Developing the basic layout: The complete layout of the IDE is mainly categorized as the following

- Horizontal Navigation Bar
- File system
 - Project div
 - Controller div
- View
 - Page view
 - Code view
- HTML tool bar

Horizontal navigation bar is the one which is similar to that of any IDE which includes file, edit, search, project and help. Each in turn has its own drop down of options. Project div is the Tree-view listing of the projects saved to the saved by the developer. Controller div is the vertical

dropdown list where in the entire collection of controller files (a kind of libraries) on the server are listed, each in turn has the list of the functions in it.

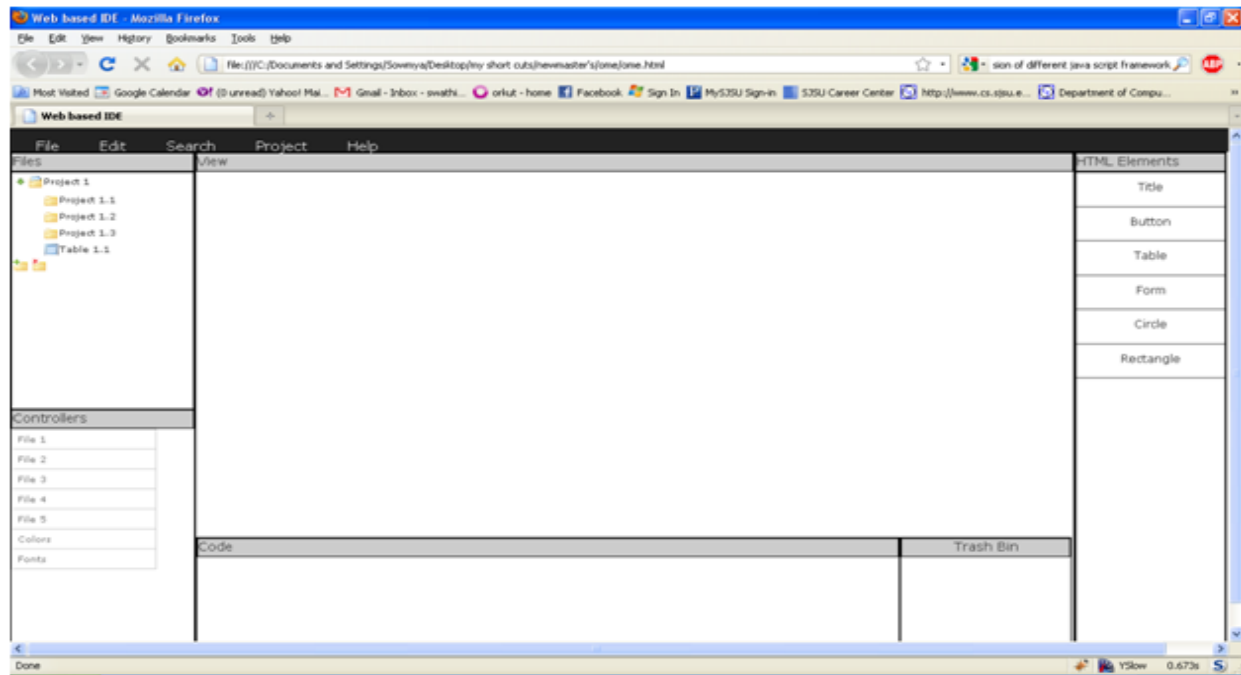


Fig. 8 Basic layout with no functionality

The developer is given option of viewing both the page view and the code view. Whenever he drops an HTML tag from the toolbar the element appears in the page view and the corresponding code will appear in the code view. HTML tool bar is the one the developer uses to create the XHTML pages by dragging these elements on to the page view or code view. Fig. 8 shows the basic layout without any functionality developed using basic HTML <div> and <u> tags and CSS.

Developing the drop downs: Each element in the horizontal navigation bar has a drop down which will be triggered by a mouseover event. For making the IDE user friendly and include the drop down functionality of the Controller files, the drop down list of the Controller files which contain only the function names will be triggered by a mouseclick event on the controller file as shown in the Fig. 9.

This is achieved by adding the following functions.

```
node.onclick=function() {
    this.className+=" over";}
node.onmouseout=function() {
    this.className=this.className.replace(" over", "");
}
```

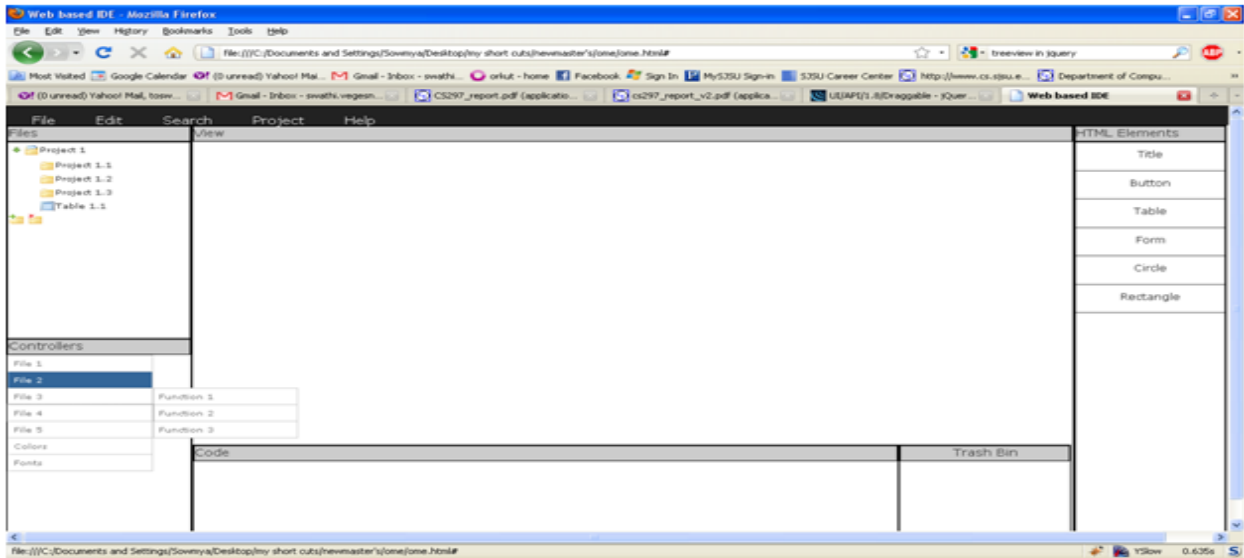


Fig. 9 Basic layout with drop downs

Creating the XHTML elements: The elements listed on the right hand side table of the IDE list all the HTML lists. On a drag and drop of these elements on the view component we can obtain the preview of the HTML document created. In order to obtain the link with the controller functions the user has to drag and drop the HTML elements on to the controller files.

A working of the basic layout is shown in the Fig. 10, which includes the title tag, a button and a table. These elements can be dropped on the controller files to obtain the link to the functions.

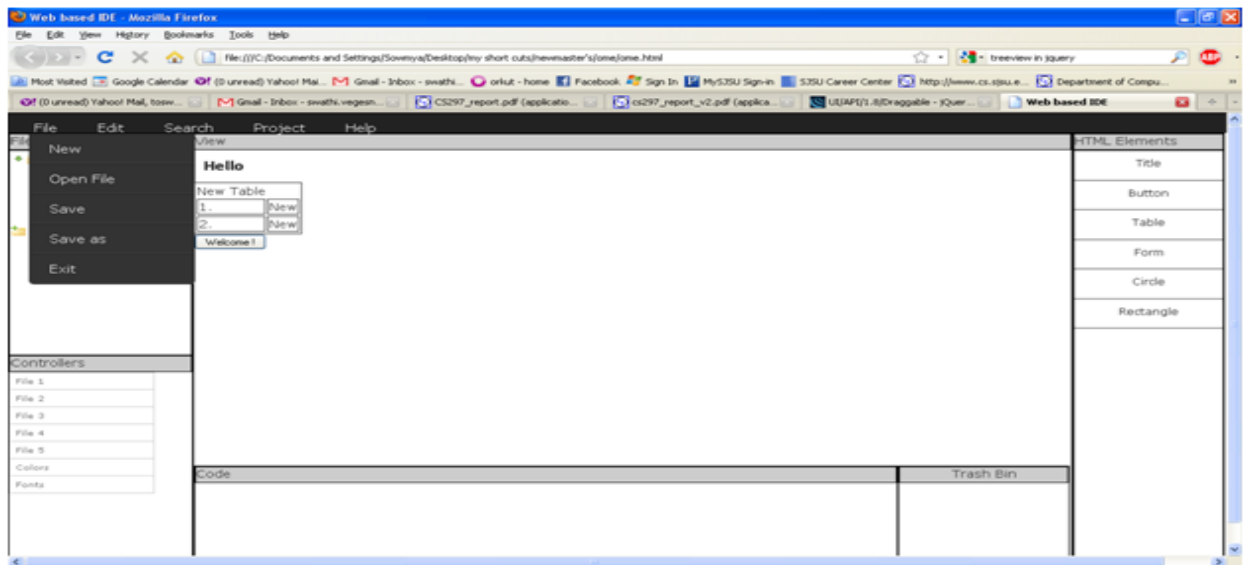


Fig. 10 Basic layout

5.4 Remarks

The Layout developed includes all the basic functionality of the drop downs and the drag and drops of the IDE. This deliverable made be accomplish the basic foundation of the IDE and gave me knowledge of the technical concepts needed to work further. All the future work on the Web-based IDE will be on the top of this layout.

6. Conclusion

The CS297 project gave me an opportunity to explore new language PHP and new frameworks like CakePHP, jQuery, YUI, DOJO and Prototype. CakePHP gave me extensive knowledge of how to work on the MVC frameworks and also aided my understanding of the controller components. This semester's four deliverables helped me to solidify the technology and the resources required to accomplish the final proposed project. This semester involved exploring various technologies and doing a proof of concept to test each technology. All four deliverables were successfully accomplished. The basic layout of the Web-based IDE has been developed as a part of CS297, with good understanding of the CakePHP framework and the features that can be implemented in the final project using jQuery. In the next semester layout of the IDE should be fully completing including saving and retrieving the dynamic pages created by the developer from the server. The future work include implementing all the drag and drop functionality to all the IDE elements to link up the view and controller components in building the dynamic web pages, setting up the server to store all the files created by the user and also have all the file structure, setting up the database tables for storing users project names and related credentials, transforming the complete IDE developed onto the CakePHP framework separating the model, controller and the view components of the IDE and conduct manual tests to test all the functionality of the elements on the IDE.

References

- [1] Drag and Drop site creator. <http://www.dragdropsitecreator.com/index.php>
- [2] CakePHP. Retrieved May 05, 2010, from <http://cakephp.org/>
- [3] Top ten JavaScript frameworks. Retrieved May 05, 2010, from <http://speckyboy.com/2008/04/01/top-10-javascript-frameworks-which-do-you-prefer/>
- [4] Animated post card in jQuery. Retrieved Feb 10, 2010 from <https://buildinternet.s3.amazonaws.com/live-tutorials/postcard/index.html>
- [5] Slickspeed. Retrieved May 05, 2010, from <http://mootools.net/slickspeed/>
- [6] Microsoft Access. Retrieved March 20, 2010, from <http://office.microsoft.com/en-us/access/ha102473131033.aspx>