



Web-based IDE for Interfacing View Controller

Presenter: Tejasvi Palvai
CS 298

Advisor- Dr. Chris Pollett
Committee Members- Dr. Mark Stamp
Dr. Robert Chun



Outline

- Purpose
- Why Web-based IDE?
- Tools
- Features
- Performance issues
- Conclusion
- References



Purpose

- The main purpose of this project is to develop a web version of an IDE i.e., Web-based IDE (Coding in the cloud) for Struts based Projects.
- To interface view and controller components of MVC architecture.
- Enables users to create XHTML pages using the drag and drop mechanism.
- Developed this using Struts framework.



Why Web-based IDE?

- An IDE is a software application that provides comprehensive facilities like a source code editor, compiler for development.
Eg: Eclipse, Net Beans.
- IDEs are desktop-based applications. Users have to install and configure these applications.
- Web-based IDEs has numerous advantages.....



Advantages of Web-based IDE

- Instant Development.
 - Avoids environment setting or configuration.
- Accessing the Code from anywhere.
 - All we need is Internet connection .
- Layout.
 - Layout techniques of web applications guide users in certain focused directions.

Existing Web-based IDE

- Bespin(Developed by Mozilla)

It is a web-based IDE (editor) that allows developers to collaborate on code-based projects.

Disadvantages:

- At present it supports only JavaScript, HTML, CSS code
- Compatible only with Firefox browser.
- JavaScript errors with IE.
- The present version has sluggishness in the cursor movement.

Tools

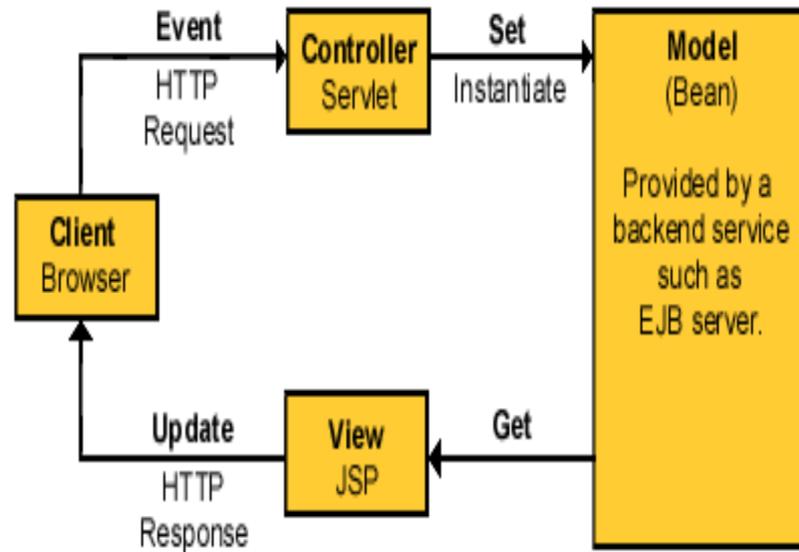
- Struts Framework
 - Open source framework based on MVC architecture.
- CKEditor
 - Open source text editor
- Tomcat Server
 - Used to host this Web-IDE.
- Firebug
 - Helps in finding errors and debugging.
- Languages
 - Java



MVC architecture

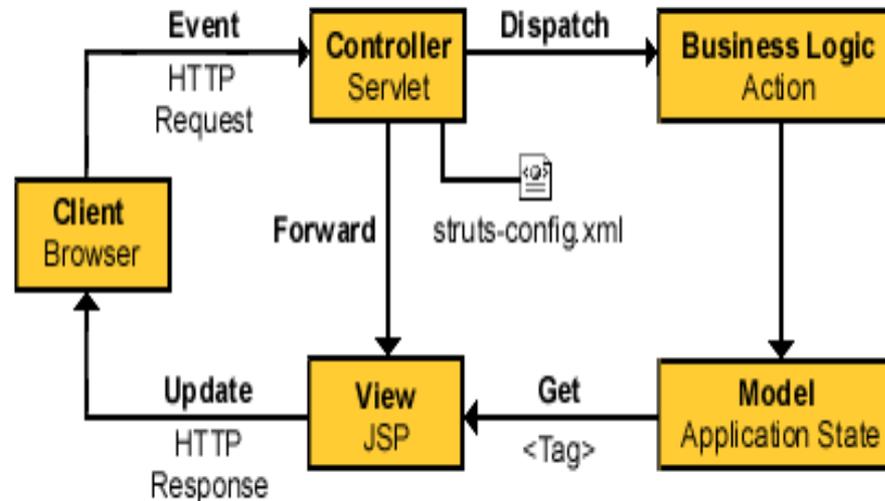
- Model-View-Controller is architectural pattern in Software Engineering.
- Separates business logic and application data from presentation data.
- Model
 - It is both data and business logic to manipulate data.
- View
 - Represents user view of application and takes inputs from user
- Controller
 - Receives input and initiates a response by making calls on model objects

MVC ctd..

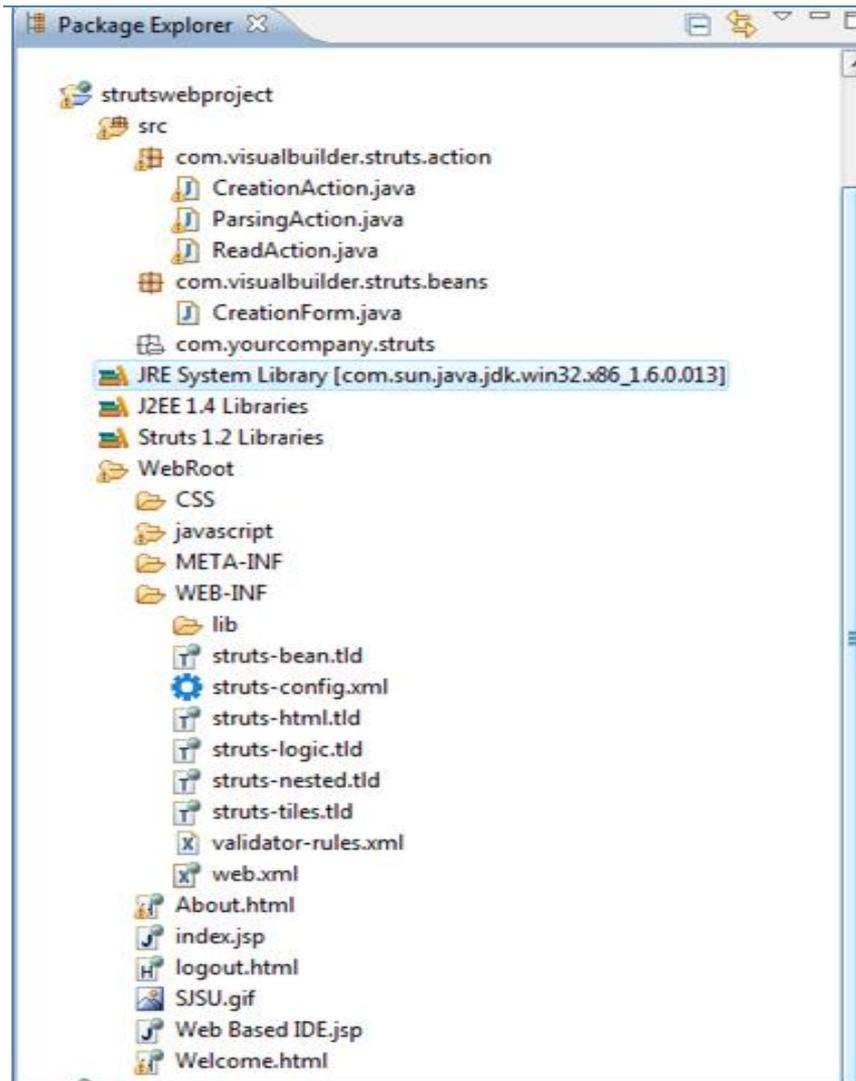


Struts

- It is an open source framework based on Model-View-Controller architecture.
- A Standard for developing well-architecture Web applications .
- Easy to integrate with client side technologies.



Directory structure for Struts based Projects





Action classes

- It is the developer's responsibility to create these classes.
- They act as bridges between user-invoked URIs and business services.
- Actions process a request and return an ActionForward object that identifies the next component to invoke.
- They are part of the Controller layer, not the Model layer.



Form files

- They act as a "firewall" between forms (Web pages) and the application (actions).
- These components allow the validation of user input before proceeding to an Action.
- In this project, these files are kept under `com.visualbuilder.struts.beans` and are typically written using Java beans.

Configuration files

- These files are mainly used for establishing the connection between web application and web container.

Struts-Config.xml:

- It is the main configuration file.
- Its where all the struts Actions are defined and what JSP pages are used to display them .

Web.xml:

- The web.xml file provides configuration and deployment information for the web components that comprise a Web application.

```
<welcome-file-list>  
  <welcome-file>Welcome.html</welcome-file>  
</welcome-file-list>
```

Preliminary Work

- Developed the basic struts application.
 - Created a web application to get comfortable with Struts
- Compared different JavaScript frameworks(YUI, JQuery,DOJO)
 - YUI,DOJO uses more memory when compared to JQuery and DOJO. YUI took around 78k and JQuery used 65 k of memory .
 - JQuery performs well in almost all the browsers and also has effective and short code.

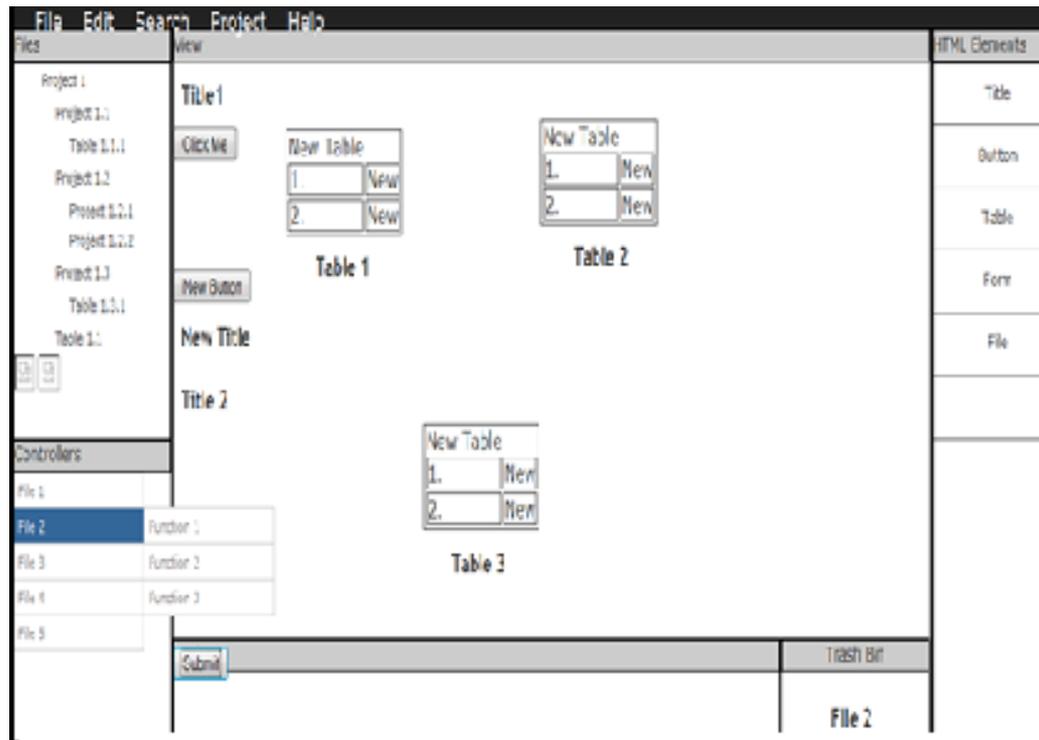


Layout for IDE

- Left part of IDE is for managing the projects.
- Top part is the horizontal menu bar. These are stacked next to each other.
- The right most part of the IDE has HTML elements to create forms.
- The central part is the edit part. Its purpose is to write and read the code.

UI of IDE

- Created the View part of the IDE



Login Page

- The project root can be accessed by using the url
url <http://localhost:8080/strutswebproject/> .
- The above url will redirect to
<http://localhost:8080/strutswebproject/WEB-INF/Welcome.html/>
internally.



The screenshot shows a login page for San José State University. At the top, there is a blue banner with the university's name on the left and a photograph of a building on the right. Below the banner, the title "Web Based IDE" is centered. Underneath the title, the instruction "Please enter your username and password" is displayed. There are two input fields: "Username" with the value "test" and "Password" with four dots. At the bottom, there are two buttons: "Submit" and "Reset".

San José State University

Web Based IDE

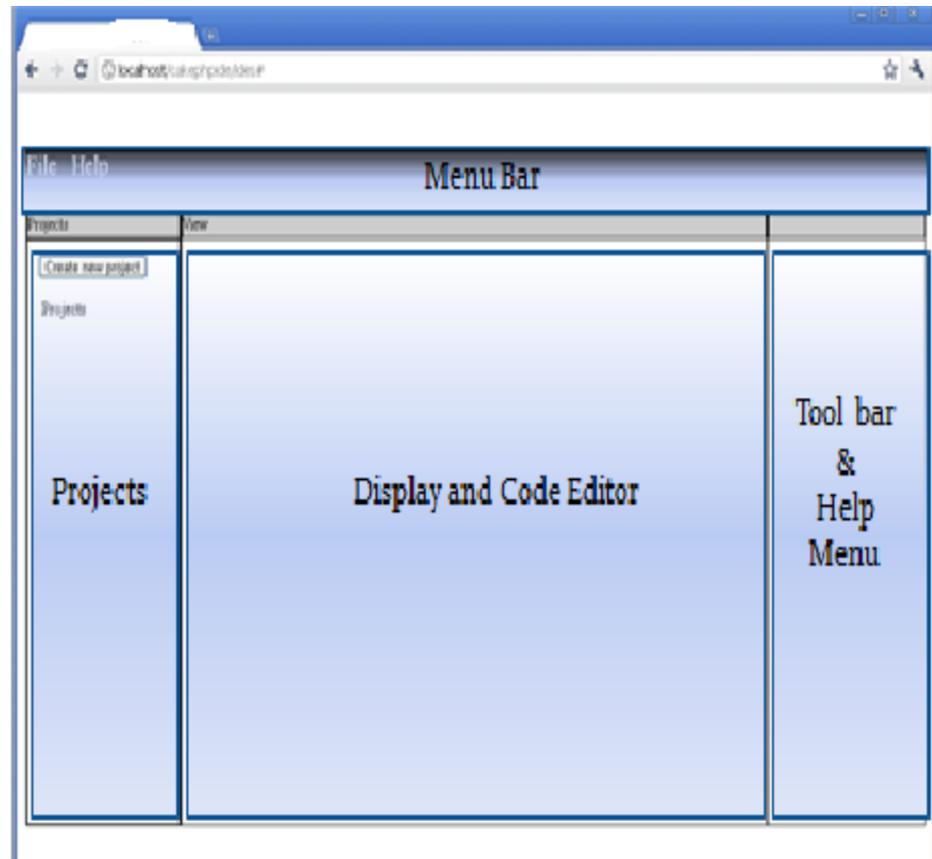
Please enter your username and password

Username

Password

Features of IDE

- Project Creation.
- Horizontal Menu.
- Tabs for View.
- Tool bar.



Projects Creation

- When a user creates a new project, 3 subfolders get created in server.
 - Action

The Action folder has the Action.java, which is a controller file.
 - Model

The Form folder has a Form.java file which is a Model file.
 - Web-Inf

It has three different files, namely Index.jsp, Struts-config.xml and web.xml files.

Implementation

- When a user creates a project, it stores in the server.
- One can get path of the server by `getServlet().getServletContext().getRealPath();`.
- The folder is traversed and sent to the view through a string buffer. `Request.setAttribute("ProjectUserName", Buffer);` method sends buffer to client side.
- In client side values can be retrieved by using `<%StringBuffervalues=(StringBuffer)request.getAttribute ("ProjectUserName "); %>`

Implementation

- Once the buffer value is retrieved, for each project an 'ul' element is created and for each file inside the folder, an 'li' element is created.
- The following is the code:

```
for(i=0, pa=0;i<arg.length && pa<patharg.length;i++,pa++)  
{  
var num =arg[i];  
var newdiv = document.createElement('li');  
newdiv.innerHTML = num;  
newNext.appendChild(newdiv);  
newdiv.setAttribute("id",patharg[pa]);  
}
```

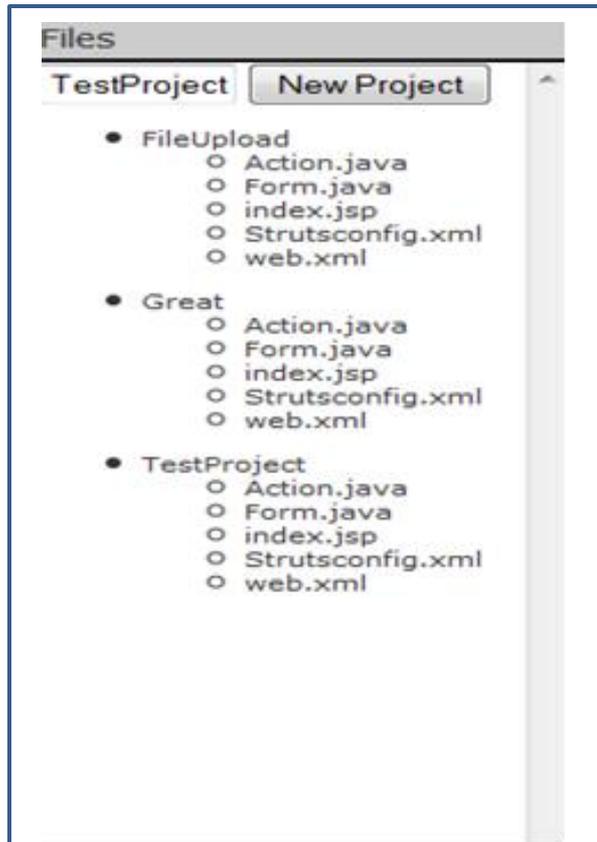
Reading and Writing to the Files

- When the user clicks on a file to read or write the code, an Ajax call is made with the pathname .
- Th java file(ReadAction.java) reads the contents of the file from server and returns the string buffer.
- When the user writes and saves the file, it gets stored.

- Following is the code:

```
$.ajax({'type' : 'POST',  
      'url' : 'Read.do',  
      data : 'project='+path_name,  
      success: function(msg)  
      {  
          msg = msg.replace(/\n/gi, "<br>");  
          $('#tab2 textarea#editor1').val(msg);  
      }  
      });  
}
```

Tree structure of user created Projects





Horizontal Menu

- The Horizontal menu bar has menu items next to each other from left to right and all other sub items are stacked vertically.
- Users can even create a new project here by clicking the 'New Project' dropdown.
- When the user clicks on new project, a modal box opens and they can enter the project name given in the place provided.
- A modal box is developed using JavaScript and is a technique for developing dialog boxes without any popups.

Implementation

- The Horizontal menu bars are implemented by using both JavaScript and HTML.

```
<ul>
```

```
  <li><a href="#" id="opencreateproject">New Project</a></li>
```

```
  <li><a onclick="return popitup('About.html')">About</a></li></ul>
```

Popup Code:

```
function popitup(url) {
```

```
  newwindow=window.open(url,'name','height=350,width=400');
```

```
  if (window.focus) {newwindow.focus()}
```

```
  return false;
```

```
}
```

Tabs for View

- Central part of Ide has two parts.
 - Design Tab
This tab is for visualizing the result of code in edit tab.
 - Edit Tab
This tab is for writing and editing the code.
- Used unordered lists for tabs.

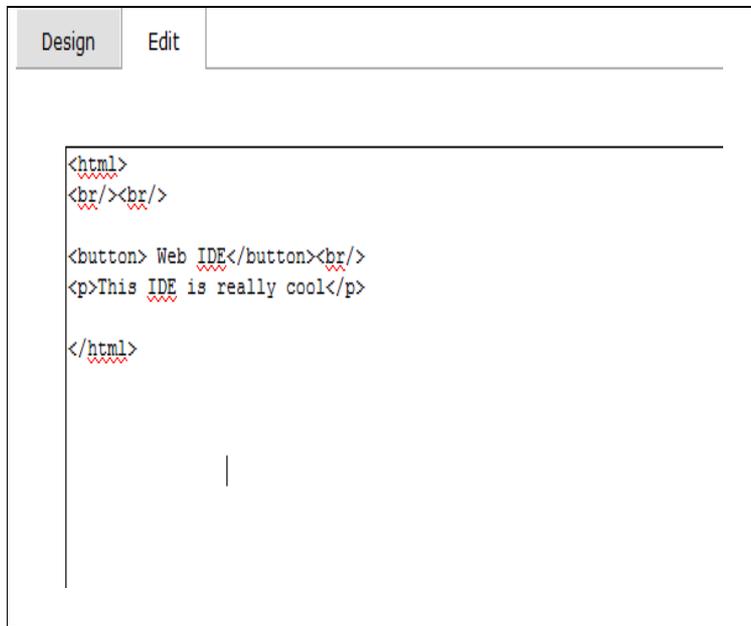
```
<ul class="tabs_view">
```

```
  <li><a id = 'design' href="#tab1">Design</a></li>
```

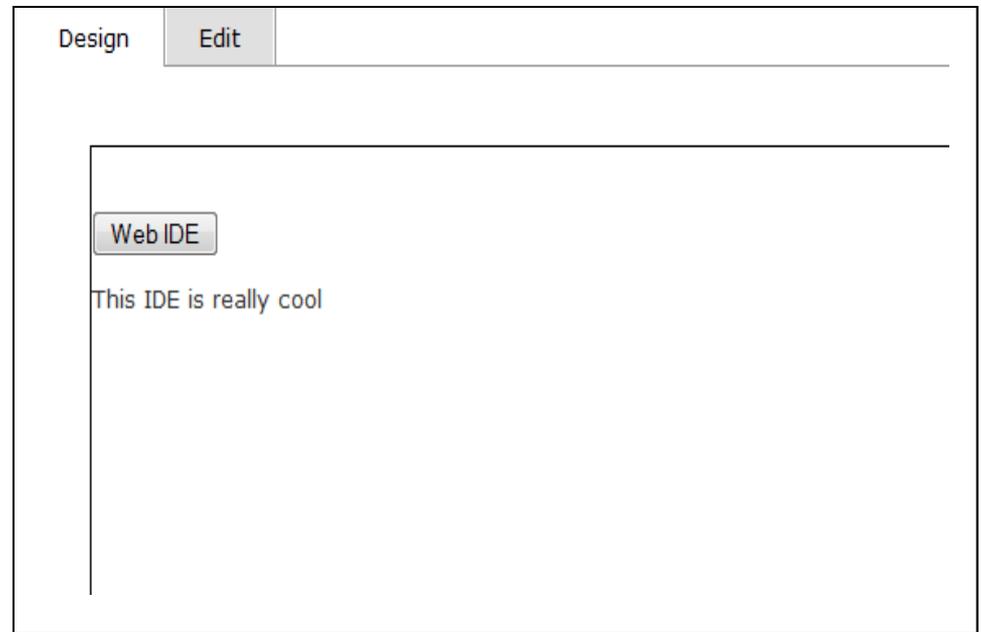
```
  <li><a id = 'edit' href="#tab2">Edit</a></li>
```

```
</ul>
```

Design and Edit Tabs



```
<html>
<br/><br/>
<button> Web IDE</button><br/>
<p>This IDE is really cool</p>
</html>
```



Implementation

Design Tab:

- When clicked on Design tabs, all contents on edit tabs is passed to a function which decodes the special characters and a String.
- This string is converted to HTML and gets appended to the id of Design Tab.
- The following is the JavaScript code:

```
$("#design"). click(function () {  
    var h = htmlspecialchars_decode($("#tab2 textarea#editor1").val());  
    h = h.replace("&nbsp;"," ");  
    $("#viewarea"). html(' ').html(h);  
});
```

Implementation

Edit Tab

- On clicking Edit tab, the html is passed to a function which replaces html strings and is added to edit tab.
- The following is the JavaScript code.

```
$("#edit").click(function() {  
    var h = htmlEntities($("#viewarea").html());  
    var h = $("#viewarea").html();  
    $("#tab2 textarea#editor1").val(h);  
});
```



Tool Bar

- The main advantage of these Tool bar is, users can create any forms just by dragging and dropping elements on to the view.
- The various elements that were used in this project are
 - Labels
 - Text Fields
 - Text Area
 - Drop Downs
 - Check boxes
 - Radio buttons

Implementation

- These elements are made draggable and droppable onto the view area by using the methods .
- Following is the code:

```
$(".form").draggable({  
    helper:'clone',  
    cursor: 'move'  
});
```
- These elements can be dropped only onto the View form.

Implementation

- Attributes like name, values to these elements can be set by Right clicking onto the element.

```
<div id="AttLabel" style = "border: 1px solid;padding: 1em;">
```

```
<p>Parameters needed for Label edit</p>
```

```
<form>
```

```
<div><span class="">Label For:</span><span><input id="AttLabelFor" type='text' name='LabelFor' value=""/></span></div>
```

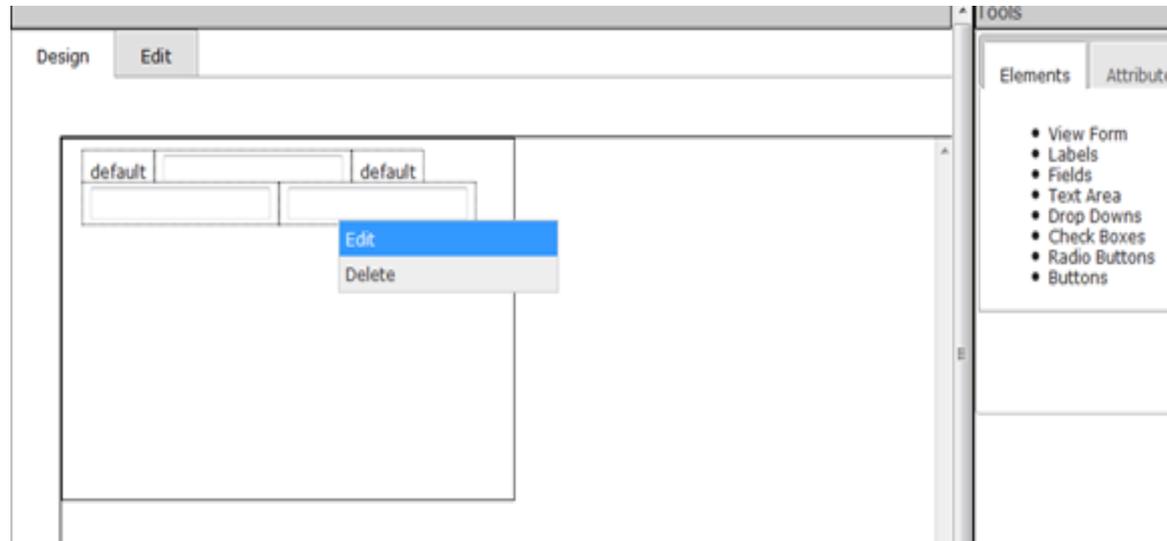
```
<div><span class="">Label Value:</span><span><input id="AttLabelValue" type='text' name='LabelName' value=""/></span></div>
```

```
<div><span class=""><input id="AttLabelAjax" type='button' name='OK' value='OK'/></span><span><input type='reset' name='Clear' value='Clear'/></span></div>
```

```
</form>
```

```
</div>
```

Attribute Setting By right click



Performance issues

- Does not have any performance issues like sluggishness in cursor movement.
- Switches very quickly between different files easily.
- Browser Compatibility
This web-based IDE is compatible with Mozilla firefox, Internet Explorer, Google Chrome and Safari .
- Challenges
One of the main challenge with these IDEs is, one must have Internet connection to have access to their code.



Performance Testing

- The performance testing of this web-based IDE and Bepin are conducted using the tool Firebug.
- Compared various features like Loading the IDE, Saving a file, Creating a file and writing to a file.
- Results showed that this web-based IDE takes less time to do all the tasks, hence more efficient than Bepin.

Performance Testing

Web-Based IDE	SJSU	Bespin
Loading a IDE	248 ms	650 ms
Creating a project	1.22 s	1.34 s
Writing to files	41 ms	710 ms
Saving a file	810 ms	1.03 s



Conclusion

- This web-based IDE increases the productivity of developers .
- Scalability, speed and productivity of this web application is comparable to desktop IDEs.
- ‘Coding in cloud’ will become mainstream IDE in near future.

References

- [1] Implementing MVC architecture using Struts
- http://www.oracle.com/technology/sample_code/tech/java/j2ee/jintdemo/tutorials/Struts.html
- [2] Struts tutorial by Benmira free
- <http://benmira.free.fr/en/j2ee/struts3.htm>
- [3] Bespin Web-based IDE
- http://www.rotorcreative.com/interactive_bespin.php#
- [4] JavaScript Frameworks:
<http://blog.creonfx.com/javascript/mootools-vs-jquery-vs-prototype-vs-yui-vs-dojo-comparison-revised>



Thank You.