

## **Comp 466 – Assignment #2**

Moshiur Howlader

January 12, 2019

Please note my Athabasca ID to login to my MySQL database is:

**ID: moshiurho**

**Password: moshiur391**

In case running the .sql file is necessary.

### **Homepage**

Hyperlink: [http://student.athabascau.ca/~moshiurho/Comp466\\_Assignment2/tma2.htm](http://student.athabascau.ca/~moshiurho/Comp466_Assignment2/tma2.htm)

The homepage was designed to contain the launching page to allow directory access to all the parts of the web applications implemented for this assignment. A clean, simplistic design was used for the overall look of the web application, with a navbar at the top for easy access to each part of the assignment. Also, a footer to cover the bottom of the webpage was implemented. Majority of the CSS design for the Web Application was implemented in this stage of the assignment. A small container containing icon hyperlinks to the two web application as well as its explanation is contained in there.

Note, the documentation for this assignment is placed in the “documentation” tab of this website.

### **Part 1:**

Hyperlink:

[http://student.athabascau.ca/~moshiurho/Comp466\\_Assignment2/part1/part1.php](http://student.athabascau.ca/~moshiurho/Comp466_Assignment2/part1/part1.php)

Part 1 required an implementation of an online bookmarking system “**BookIt**” which allows users to create an account, store or manage their bookmarks, as well as allowing the users to logout of the system. Before logging in the app shows the user the (up to) top 10 most bookmarked links in the system. To implement this system, I made two MySQL tables to store the appropriate data for the system:

```

CREATE TABLE userList(
    username varchar(255) NOT NULL,
    passwd varchar(255) NOT NULL,
    PRIMARY KEY (username)
);

CREATE TABLE bookmark(
    LinkId int NOT NULL AUTO_INCREMENT,
    urlLink varchar(255) NOT NULL,
    username varchar(255),
    PRIMARY KEY (LinkId),
    FOREIGN KEY (username) REFERENCES userList(username)
);

```

The table “userList” has a primary key of “username” to store unique userId in the system, as well as the password associated to that system. The table “bookmark” is used to store the linkId with a primary key to make it a unique entry into the system. A foreign key to associate the “username” in table “bookmark” to “userList” was made. Finally, the actual bookmarked link is stored in the database.

There were several PHP files used to service this app. loadTop10Link.php was used to load the top 10 links when user first loaded the web page. loadTop10Link\_DOM.php would recreate the previous script when the web app would recreate the landing/homepage through Javascript DOM. login.php and signup.php, as the name implies does the appropriate bookkeeping to login or signup users into the BookIt system. Finally, the addDeleteEditCheck\_Links.php script manages the main user experience tasks such as adding, deleting, editing, or checking the links the user attempt to work with in the BookIt system while they are logged in. Various Ajax requests from part1.js as well as JS DOM to make the app work.

To view this application, simply click the “Part 1” tab in the Navbar of the Web application to navigate to the hyperlink.

## **Part 2:**

Hyperlink:

[http://student.athabascau.ca/~moshiurho/Comp466\\_Assignment2/part2/part2.php](http://student.athabascau.ca/~moshiurho/Comp466_Assignment2/part2/part2.php)

A small-scale online learning management system that can be used to deliver online courses to learners on the web called “**Educord**” was requested to be implemented. There are two user types supported. They are “instructors” and “students” user types. Before a user can login, there is an Ajax powered drop down menu to allow users to preview the courses uploaded in the Educord system. Once an instructor is logged in, they can view announcements that they made, upload their EML zip folder, view list of students enrolled in the class, assignment submissions made by their students, quiz marks for the quizzes they have assigned, and select courses that they teach to switch and to work with that course. Note an instructor can teach multiple courses.

When a student is logged in, they can view an announcement for the selected course, view and choose lecture material based on week, view relevant reference links posted by the instructor, view and submit assignment/assignment submissions, attempt quizzes posted by the instructor, and choose/enroll/withdraw from courses. The below UI should make the app usage clear:

Figure: Landing Page of Educord allowing users to select and preview Educord courses.

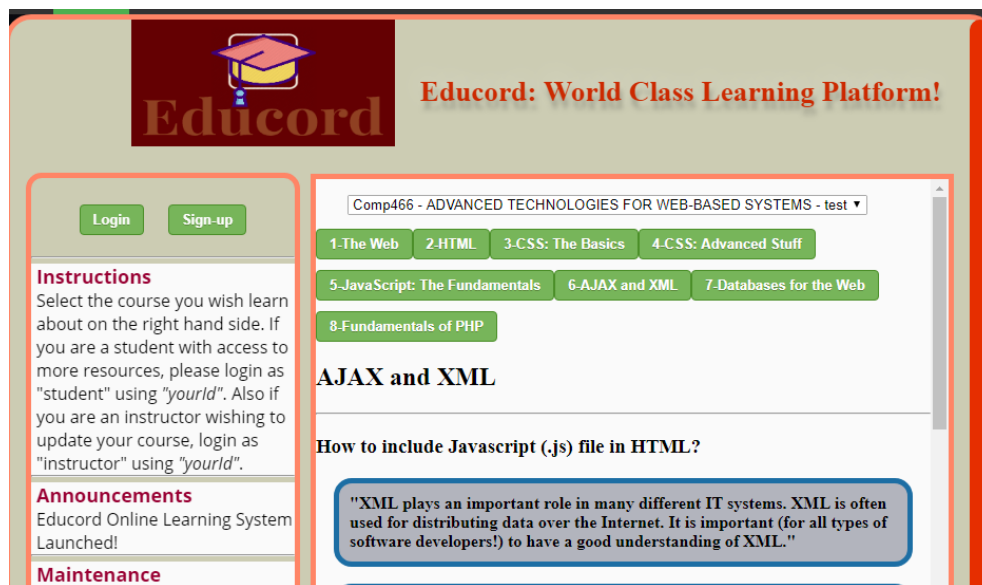


Figure: User Interface for Instructors

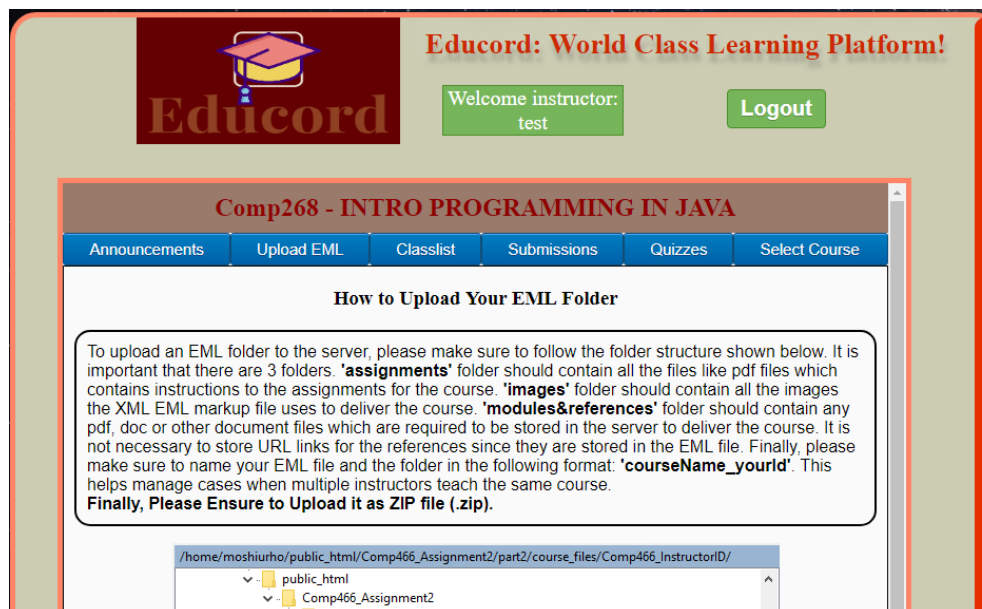
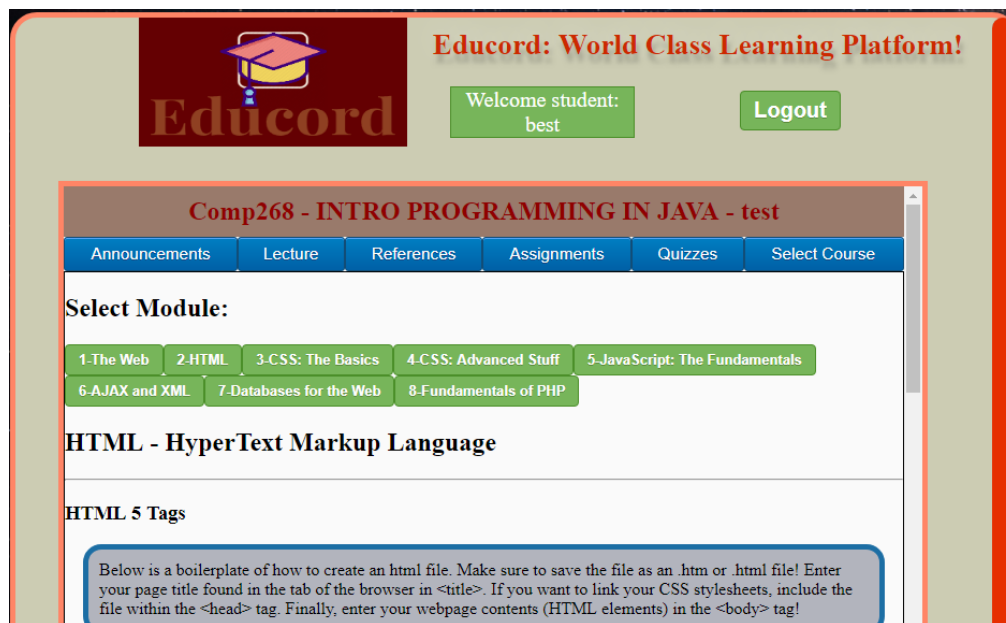
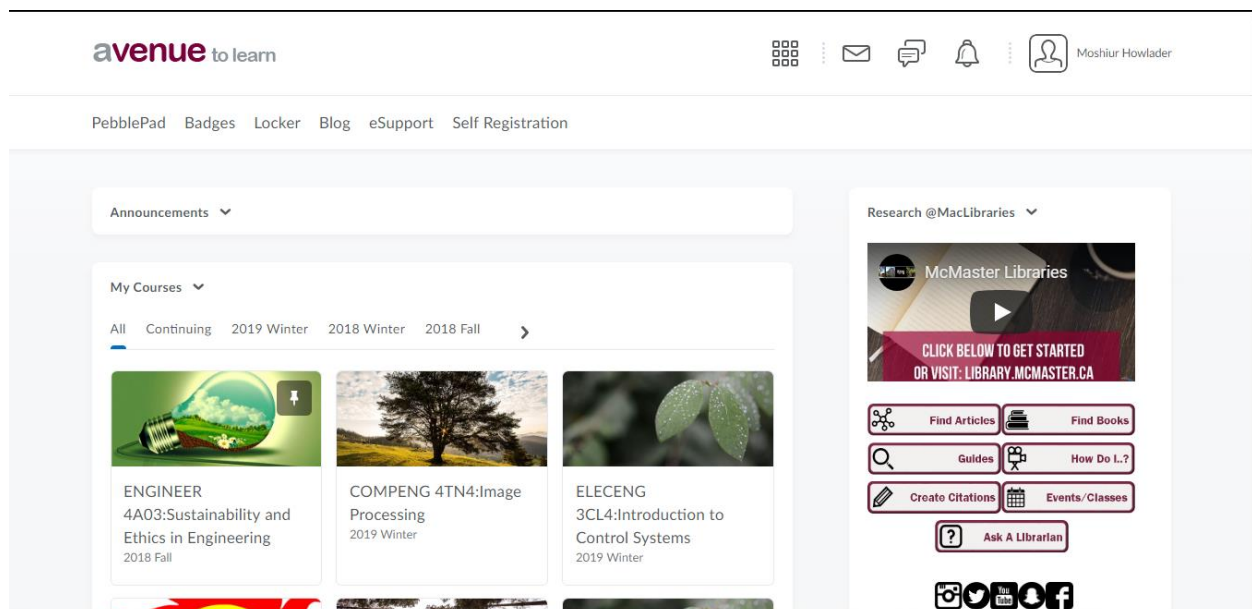


Figure: User Interface for Students



## System Analysis and Design

Majority of the design inspiration came from Avenue To Learn, McMaster's online learning management system:



As well, from the design from the research paper: “XML and Databases for E-Learning Applications” by H.F. El-Sofany, S.A. El-Seoud, F.F.M. Ghaleb, S.S. Daoud, J.M. ALJa'am and A.M. Hasnah.

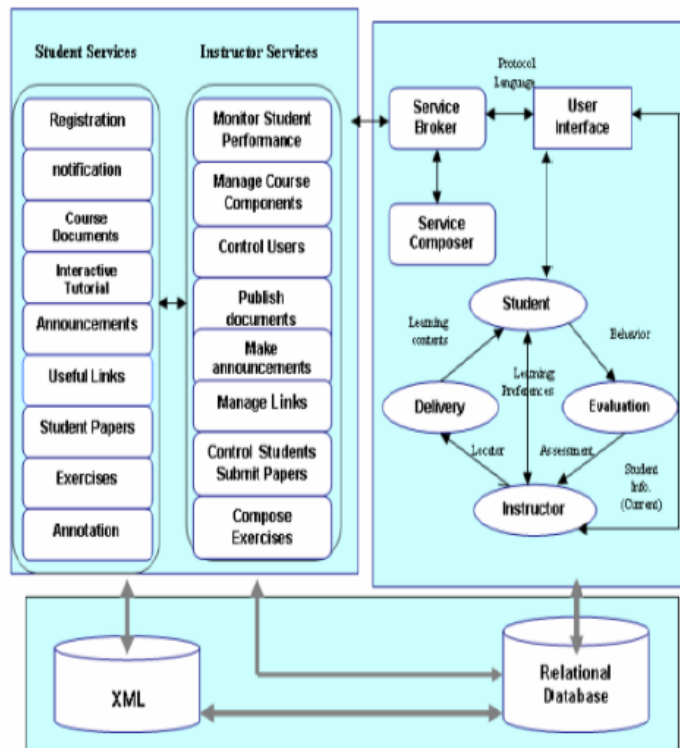


Figure7. Framework for Our Web-Based e-learning system

Instead of what is shown in Figure above, my functional modules for my system are:

	<b>Instructor Capabilities</b>	<b>Student Capabilities</b>
1	Upload EML Zip Folder which is used to populate the course contents	Enroll/withdraw in courses from a list of uploaded Educord courses
2	View Announcements made for a given course	View Announcements made for a selected course
3	View Classlist for a given course	View Lecture Modules one at a time
4	View Assignment Submissions by students	View Helpful Reference link posted by the instructor
5	View Quiz Marks by students	View/Submit Assignments as well as attempt Quizzes posted by the instructor

The instructor must first upload the EML Zip folder in a properly formatted format as shown below:

Name	Size	Changed	Rights	Owner
..		2019-01-02 9:44:32 PM	rw-rw-rw-	moshiur...
assignments		2019-01-02 12:55:49 AM	rw-r--r--	apache
images		2019-01-02 12:55:49 AM	rw-r--r--	apache
modules&references		2019-01-02 12:55:49 AM	rw-r--r--	apache
Comp466_InstructorID.xml	49 KB	2019-01-02 9:38:34 PM	rw-r--r--	apache

To upload an EML folder to the server, the instructor must make sure to follow the folder structure shown above. There are to be 3 folders. **'assignments'** folder should contain all the files like pdf files which contains instructions to the assignments for the course. **'images'** folder should contain all the images the XML EML markup file uses to deliver the course.

**'modules&references'** folder should contain any pdf, doc or other document files which are required to be stored in the server to deliver the lecture component of the course. It is not necessary to store URL links for the references since they are stored in the EML file. Finally, the instructors should make sure to name their EML file and the folder in the following format:

**'courseName\_yourId'**. **courseName** is the course code of the course they teach, and **yourId** is the id the instructor uses to login to Educord. This helps manage cases when multiple instructors teach the same course. Finally, Please Ensure to Upload it as ZIP file (.zip).

Figure: The upload UI for instructors. All a button for erasing all files within the “course\_files” exists for ease of testing the system by the tutor (in case they make mistake on uploading the file contents).

#### Upload EML Zip Folder Below:

Choose File No file chosen

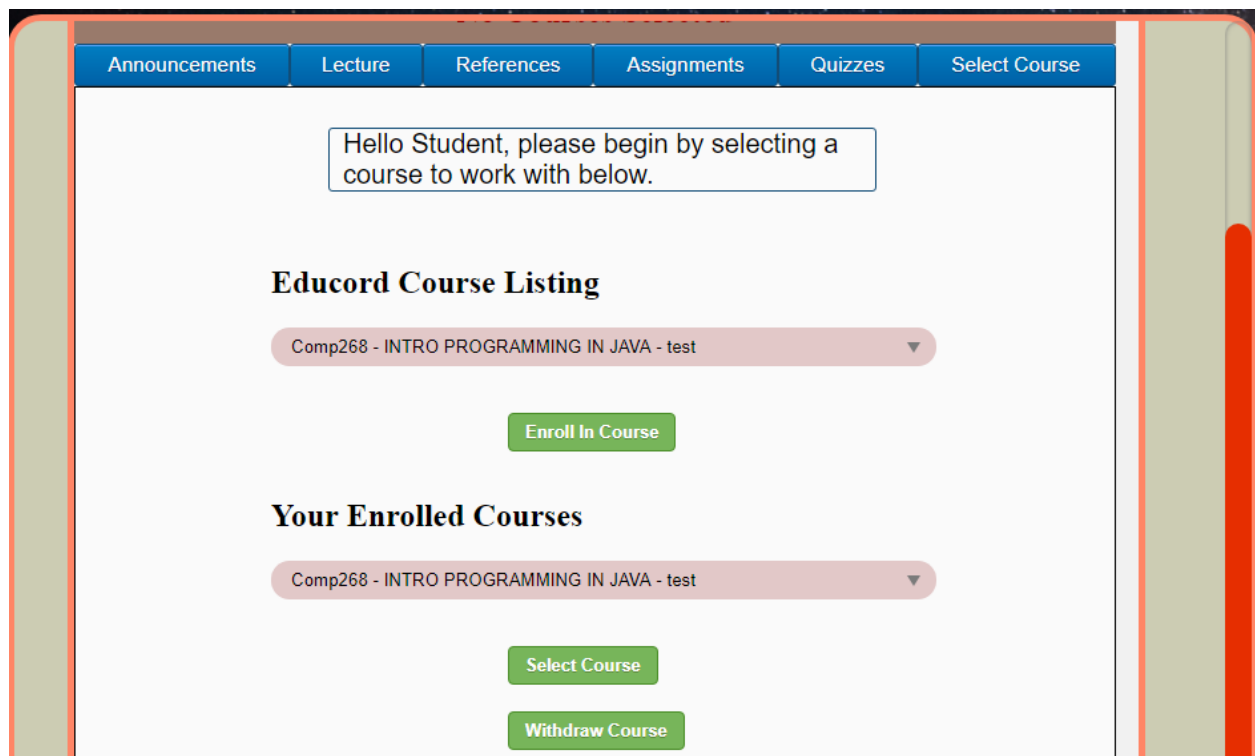
Save EML

#### Delete All EML Folders from Server (doesn't affect database contents):

Delete All EML

Once the instructor uploads the EML ZIP folder, they can select their uploaded course under the tab “**Select Course**”. Once they select the course, all the other capabilities (2 to 5) of an instructor is enabled by navigating to the appropriate tab.

From the student perspective, it is easy to use the Educord System. All they have to do is go to “**Select Course**”. Enroll in a course that is being offered in Educord, then simply select the **green** “**Select Course**” button. Once they do that, they can view, and use all capabilities available for the student perspective of the Educord system.



### Educational Markup Language

Here I will present the EML I have designed for this project.

Tag	Usage	Example
<course>	To initialize Educord EML.	<pre>&lt;course courseCode="Comp466" courseType="Computer Science" courseName="ADVANCED TECHNOLOGIES FOR WEB-BASED SYSTEMS"&gt;&lt;/course&gt;</pre>
	<p>courseCode: define coursecode, must be contiguous string</p> <p>courseType: define course type</p>	

	courseName: define name of the course	
<announcements>	To initialize the Announcement sections the students will see	<announcements> </announcements>
<announce>	<p>To make an announcement for a given date. Supports insertion of document tags to render in announcements</p> <p>date: specify the date in YYYY-MM-DD format</p>	<pre>&lt;announce date="2018-09-05"&gt;     Week 1 - The Web contents uploaded. Please     review your course syllabus and review Week 1 contents     please!     &lt;document src="Comp466_Syllabus.pdf"&gt;Comp     466 Syllabus&lt;/document&gt; &lt;/announce&gt;</pre>
<modules>	To initialize the Lecture sections the students will see	<modules> </modules>
<module>	<p>To populate lecture content for a given module. Support following tags: heading, subheading, line, video, image, desc, code, table (with row, and col), and bulletpoints (with point). All assets used to render the document must be stored in the <b><u>"modules&amp;references"</u></b> section of the EML zip folder.</p> <p>week: specify the week this module was covered in semester</p> <p>title: title or name of the module</p>	<pre>&lt;module week="1" title="The Web"&gt;     &lt;heading&gt;History of the     Internet/Web&lt;/heading&gt;     &lt;line&gt;&lt;/line&gt; &lt;/module&gt;</pre>
<heading>	Use to create headings for a module section	<heading>HTML - HyperText Markup Language</heading>
<subheading >	Use to create subheadings for a module section	<subheading>HTML 5 Tags</subheading>
<video>	Insert Youtube Videos only.	<pre>&lt;video src="https://www.youtube.com/watch?v=eYkXD_cGUYU"&gt;The Web 2.0&lt;/video&gt;</pre>



<image>	Insert images uploaded into the <b><u>"images"</u></b> folder of the EML zip folder	<code>&lt;image src="CSS_syntax.png"&gt;/image&gt;</code>
<desc>	Main tag for instructor to markup descriptions for their lectures	<code>&lt;desc&gt;Below is a boilerplate of how to create an html file. Make sure to save the file as an .htm or .html file! Enter your page title found in the tab of the browser in &lt;title&gt;;. If you want to link your CSS stylesheets, include the file within the &lt;head&gt;; tag. Finally, enter your webpage contents (HTML elements) in the &lt;body&gt;; tag!&lt;/desc&gt;</code>
<code>	To display code. Currently only supports HTML code, and codes that delimit with ;	<code>&lt;code&gt;       &lt;form&gt;;       &lt;input       type="insertTypeHere"...other parameters&gt;;       &lt;br&gt;;       &lt;input       type="insertTypeHere"...other parameters&gt;;       &lt;br&gt;;       &lt;input       type="insertTypeHere"...other parameters&gt;;       &lt;br&gt;;       &lt;/form&gt;;     &lt;/code&gt;</code>
<table>, <row>, <col>	Display tables for displaying tabular data	<code>&lt;table&gt;       &lt;row&gt;         &lt;col&gt; My colum data1 &lt;/col&gt;         &lt;col&gt; My colum data2 &lt;/col&gt;         &lt;col&gt; My colum data1 &lt;/col&gt;       &lt;/row&gt;     &lt;/table&gt;</code>
<bulletpoints>, <point>	For listing items in bulletpoints	<code>&lt;bulletPoints&gt;       &lt;point&gt;My Very Important Point to make &lt;/point&gt;       &lt;point&gt;My Very Important Point to make &lt;/point&gt;       &lt;point&gt;My Very Important Point to make &lt;/point&gt;     &lt;/bulletPoints&gt;</code>
<references>	To initialize the Reference sections the students will see	<code>&lt;reference&gt;       &lt;reference&gt;...&lt;/reference&gt;     &lt;/references&gt;</code>
<reference>	Populate a reference. The src for the reference can be a link or a document stored in the <b><u>"modules&amp;references"</u></b> section of the EML zip folder.	<code>&lt;reference src="https://www.w3schools.com/xml/"       week="6" title="AJAX and XML"&gt;XML     Reference&lt;/reference&gt;      &lt;reference src="mySQL_cheatsheet.pdf" week="7"       title="Databases for the Web"&gt;MySQL     CheatSheet&lt;/reference&gt;</code>

<assignments>	To initialize the Assignment sections the students will see	<pre> &lt;assignments&gt;   &lt;assignment&gt;...&lt;/assignment&gt; &lt;/assignments&gt; </pre>
<assignment>	Populate an assignment. Note the actual assignment document must be stored in the <b>"assignments"</b> of the EML zip folder.	<pre> &lt;assignment src="Assignment1_Comp466.pdf" due="2018-10-06" week="1,2,3,4"&gt;Assignment#1-HTML and CSS&lt;/assignment&gt; </pre>
<quizzes>, <quiz>, <question>, <choiceA>, <choiceB>, <choiceC>, <choice>, <answer>	Tags used to populate quiz contents. They produce 4 choice multiple choice question. The attributes and contents to populate are self-explanatory. Only one answer allowed. Must have 4 choices.	<pre> &lt;quizzes&gt;   &lt;quiz week="1,2,3,4" title="HTML and CSS Quiz"&gt;     &lt;question text="What does HTML stand for?"&gt;       &lt;choiceA&gt;High Transmisson Mail Letter&lt;/choiceA&gt;       &lt;choiceB&gt;High Transmission Message Line&lt;/choiceB&gt;       &lt;choiceC&gt;Hypertext Markup Language&lt;/choiceC&gt;       &lt;choiceD&gt;Hypertext Message Link&lt;/choiceD&gt;       &lt;answer&gt;choiceC&lt;/answer&gt;     &lt;/question&gt;   &lt;/quiz&gt; &lt;/quizzes&gt; </pre>

### Database Design

<u>Table</u>	<u>Purpose</u>
educordUserList	Store user (instructors and students) of the Educord System
educordAnnouncements	Store announcements of all courses in the Educord System
educordModules	Store module/lecture contents of all courses in the Educord System
educordReferences	Store references of all courses in the Educord System
educordAssignments	Store assignments of all courses in the Educord System
educordQuizQuestions	Store quiz questions of all courses in the Educord System
educordCourseContent	Store a list of all courses uploaded in the Educord System
educordEnrolledStudents	Store list of students enrolled in various courses in the Educord System
educordAssignmentSubmissions	Store information about student submissions of assignments in the <b>"submission"</b> folder of the Educord System
educordQuizGrades	Store quiz grades of all courses in the Educord System

### Parser/PHP files Explained

The parser used to translate the contents from EML to a nicely formatted, easy to retrieve EML format to store into the database was done by the file “**uploadEML.php**”. This php file does about half of the parsing (such as removing extra spaces, extracting the internal contents of the main tags like modules, assignments, references tags etc.). Finally, the last half of the parsing needed from the PHP was done in the file “**educordParser.php**”. This script takes the appropriate Ajax requests from **Part2.js** and returns the HTML string code back to the client. This HTML string code is HTML code in string format. This string formatted HTML code was translated into actual DOM elements on client side using the HTML5 feature of **templates**. The remainder of the script are self-explanatory scripts to allow Educord to signup users, log users in, delete EML zip folders on server, delete server-side assignment uploaded files (for tutor's convenience), and last the script to allow user to select courses (which could have been included into educordParser.php if needed).