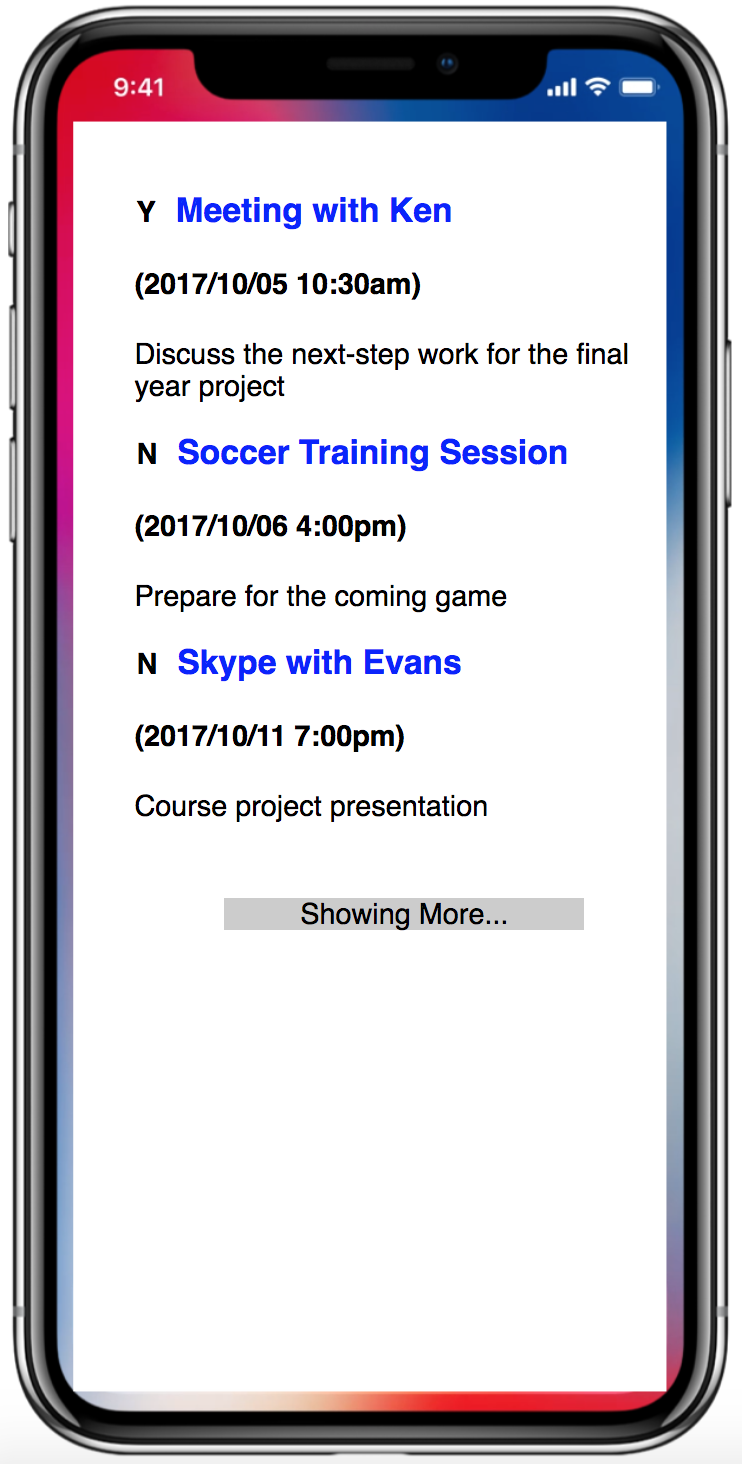
**COMP3322 MODERN TECHNOLOGIES ON WORLD WIDE WEB**

**Lab 4 PHP, JavaScript and AJAX**

**Overview**

In this lab, we will develop a simple web-based productivity application, iNote, with which we can maintain a simple to-do checklist in an iPhone interface. As shown in Fig. 1, each entry in the iPhone interface is a note entry, including the title, description and starting date/time of a task, as well as a label telling if the task has been done or not. We will practice loading notes from the database, and toggling between done and to-do status through server-side implementation (PHP) and the client-side implementation (HTML, JavaScript, AJAX). We adopt MySQL as the back-end database server to store the notes.



**Figure 1 iNote interface**

**Lab Exercise 1 - Create Checklist**

We first create a simple web page displaying a figure of iPhone X, inside which initially 3 notes are loaded. Three more notes will be loaded whenever the “Showing More” button is clicked. New notes should be appended to the end of the already loaded notes, without reloading the entire web page.

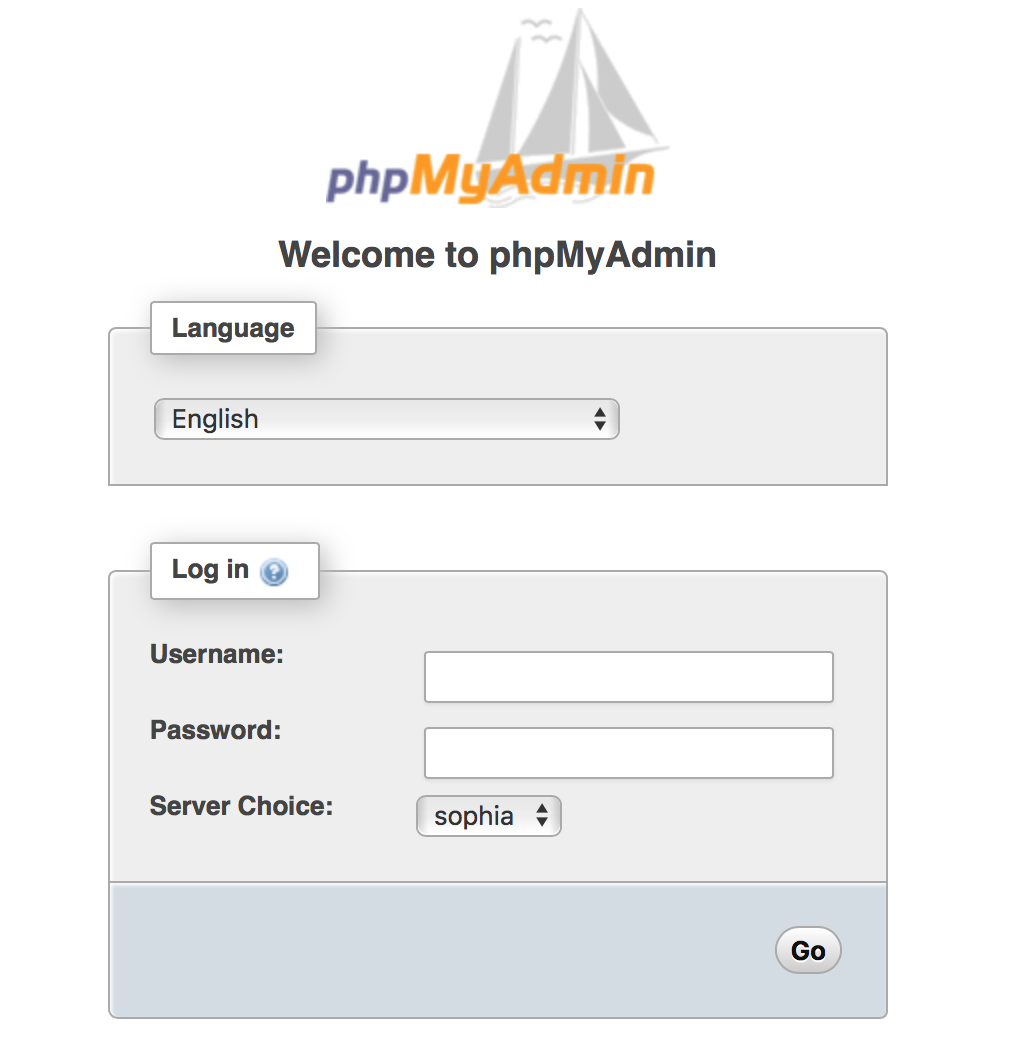
**Task 1**

Prepare the Database.

**Step1:** Get your MySQL account, if you do not have one, at <https://intranet.cs.hku.hk/common/mysqlacct/register.php>. It takes **one** working day for CS technical office to activate your account.

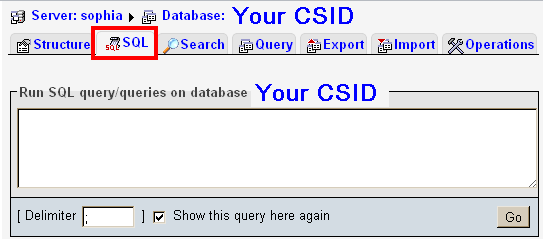
**Step2:** Log in to MySQL using phpMyAdmin.

1. Go to <http://i.cs.hku.hk/tools/phpMyAdmin/>.
2. Log in using your MySQL account, and select the server, sophia. We will use the MySQL database hosted in sophia.cs.hku.hk.



**Step3:** Select your database. After login, select your database on the left column. Your database should be named as your CSID.

**Step4:** Create tables and records. Click the **SQL** tab and execute the SQL statements for creating necessary tables and inserting data for this lab exercise.

****

|  |
| --- |
| create table checklist (  id int not null primary key auto\_increment,  title varchar(256),  datetime varchar(256),  doneOrNot varchar(1) not null default 'N',  taskdescription text  );  insert into checklist values(1,'Meeting with Ken', '2017/10/05 10:30am', 'Y', 'Discuss the next-step work for the final year project');  insert into checklist values(2,'Soccer Training Session', '2017/10/06 4:00pm', 'N', 'Prepare for the coming game');  insert into checklist values(3,'Skype with Evans', '2017/10/11 7:00pm', 'N', 'Course project presentation');  ………… |

**Task 2**

Create **index.html** as follows, in which the body contains one **<div>**, with id “Checklist”, which further consists of a **<div>** containing notes, and the **<div>** for “showing more” at the bottom. Please include the given CSS “style.css” inside the <head>.

|  |
| --- |
| <html>  <head>  <title>iNote</title>  <link rel="stylesheet" type="text/css" href="style.css">  </head>  <body>  <div id="Checklist">  <div id="Note">  </div>  <div id="more" onclick="pullMore()">  <p> Showing More...</p>  </div>  </div>  </body>  </html> |

**Task 3**

**Step 1:** In **index.html**, identify the events that initiate the AJAX communication.

1. <**body**> is first loaded. **Event: onload.**
2. <**div** id="**more**"> is clicked. **Event : onclick.**

**Step 2:** Define the event handler (a JavaScript function **pullMore()**). In the event handler, create an **XMLHttpRequest** object.

|  |
| --- |
| <script>  **lastRecord** = 0;  **function** **pullMore**(){  var **xmlhttp**;  if (window.**XMLHttpRequest**){ // code for IE7+, Firefox, Chrome,etc.  **xmlhttp** = new **XMLHttpRequest**();  }else{ *// code for IE6, IE5*  **xmlhttp** = new **ActiveXObject**("**Microsoft.XMLHTTP**");  }  //More codes here for the next step.  **lastRecord** +=3;  }  </script> |

**Step 3:** Define the response actions when the server’s response is received.

|  |
| --- |
| **xmlhttp**.**onreadystatechange** = **function**(){  if (**xmlhttp**.**readyState** == **4** && **xmlhttp**.**status** ==**200**){  // The client’s action when the response is ready.  // **Your task:** Add your code here  }  } |

**Hints:**

To refer to the HTML content of a division, you can use

|  |
| --- |
| document.**getElementById**("**division-id**").**innerHTML** |

To get the server’s response text, use:

|  |
| --- |
| **xmlhttp.responseText;** |

**Step 4:** Define the request that is sent “behind the scenes”:

|  |
| --- |
| **xmlhttp.open("GET","queryNote.php? lastRecord ="+lastRecord, true);** |

**Step 5:** Send the request.

|  |
| --- |
| **xmlhttp.send();** |

**Task 4**

Create **queryNote.php** to load new notes from the database upon client’s request. **$\_GET['lastRecord']**records the index of the last note that has been loaded.

**Step1:** Connect to database.

|  |
| --- |
| <?php  //Connect to database  **$conn=mysqli\_connect**(**'database-server'**, **'user-name'**, **'password'**) or **die**('Error! '. **mysqli\_error**($conn));  ?> |

**Step2:** Select database.

|  |
| --- |
| //Select database  **mysqli\_select\_db**($conn, **'your-db'**) or **die**('Error! '. **mysqli\_error**($conn)); |

**Step 3:** Create SQL query.We will retrieve 3 more notes.

|  |
| --- |
| //Construct your SQL query here  $**query** = '**SELECT \* FROM checklist LIMIT** '.$**\_GET**['**lastRecord**'].'**, 3**'; |

**Step 4:** Execute the query.

|  |
| --- |
| //Execute SQL query  $**result** = **mysqli\_query**($conn, $**query**) or **die**('Error! '. **mysql\_error**($conn)); |

**Step 5:** Fetch the results and generate HTML codes that display the 3 notes.

|  |
| --- |
| while($row = **mysqli\_fetch\_array**($result)) {  print "<div class=\"note\" id=".$row['id'].">";  print " …" //Add code to display the notes  print "</div>";  } |

**Hints:**

You can organize each note as <span></span><h3></h3><br><h4></h4><p></p>. The **<span>** section is used to show whether this task has been done: “Y” stands for “done”, while “N” denotes “to-do”. <**h3>** is used to show the note title, <**h4>** is used to show the starting date and time of this task, and <**p>** is used to show the task description.

**Lab Exercise 2 - Toggle Done/To-do**

In front of each note displayed, there is a flag indicating whether the task has been done (“Y” stands for “done”, while “N” denotes “to-do”). Tapping the flag will change the status from one to the other, and initiate AJAX communication to the server for changing the attribute “doneOrNot” of this note record in the database table, without reloading the entire web page.

**Task 1**

Add an event listener to the **<span>** section of each loaded note in index.html. Create a JavaScript function changeState(elem) as the event handler. Follow the steps in Task 3 of Lab Exercise 1, to implement the AJAX code for asynchronous communication with the server.

|  |
| --- |
| <script>  function changeState(elem) {  var oldValue = elem.innerHTML;  var noteID = elem.parentNode.getAttribute('id');  if (oldValue == 'Y') {  newvalue = 'N';  } else {  newvalue = 'Y';  }  var xmlhttp;  // Add AJAX code here  ………………………….. |

**Task 2**

Create **updateNoteState.php** to implement the server-side logic of changing the “doneOrNot” attribute of the corresponding note record in the database. The following code is given for your reference.

|  |
| --- |
| $value = $\_GET['newValue'];  $query = "update checklist set doneOrNot= '$value' where id=".$\_GET['id'];  mysqli\_query($conn, $query) or die ('Query Error! '. mysqli\_error($conn));  ……………………. |

**Submission**

Please finish this lab assignment before 23:59 Monday Oct 9, 2017. Please upload all files (**index.html, queryNote.php, updateNoteState.php, style.css, and theme.png**) to **i.cs.hku.hk** web server under “**lab3**”, similar to what you did in previous labs. The URL to access your page should be **http://i.cs.hku.hk/~[your\_CSID]/lab3/index.html**. We will check the page for your lab 3 marking.

*Note*: in this lab, you do not need to worry about styling of the page and text going beyond iPhone image boundary when you scroll the page. If all requirements on correctly loading notes and toggling between ‘Y’ and ‘N’ are fulfilled, you will get the "1" mark for this lab exercise. Be sure to check the page yourself after uploading it to the web server.