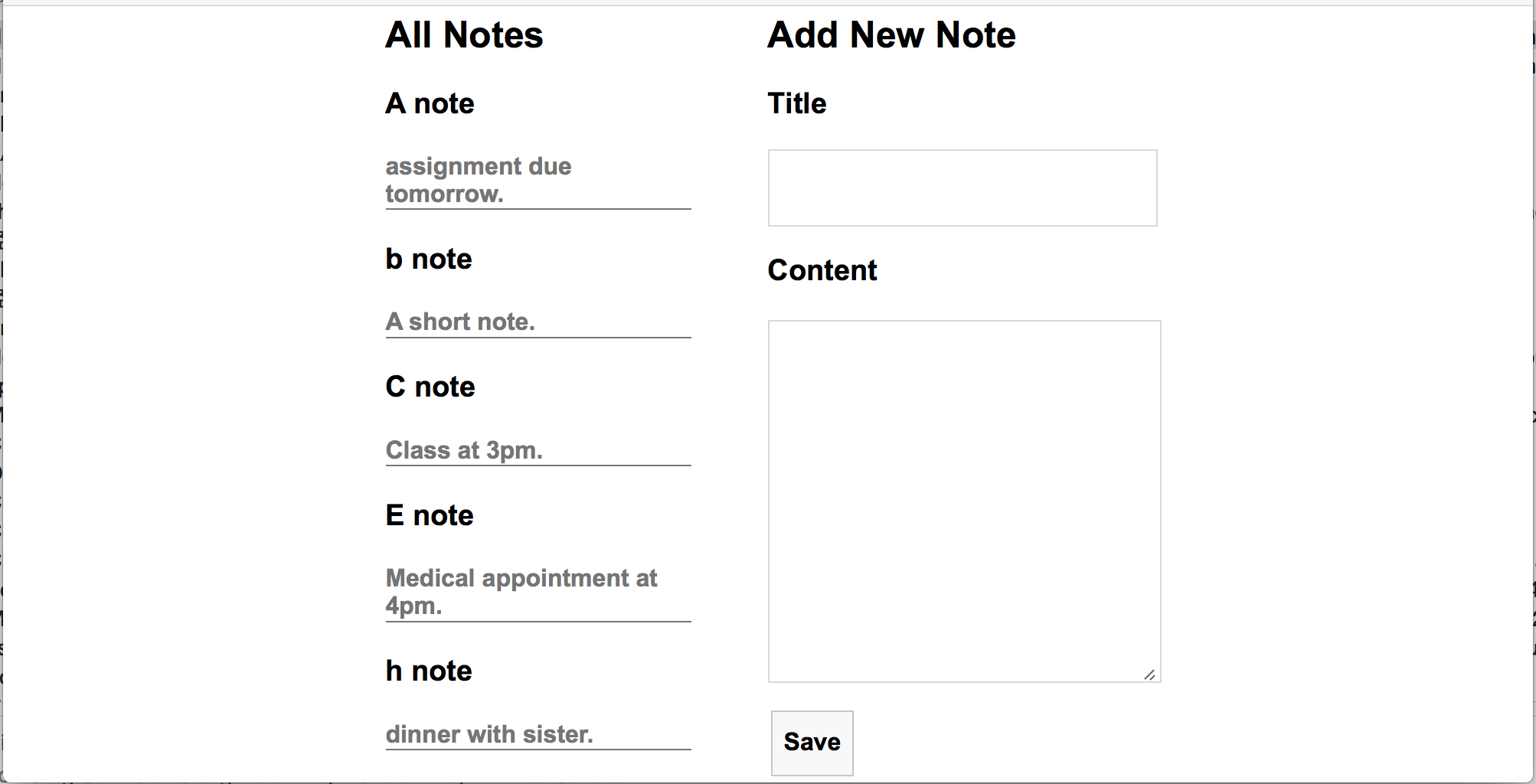
**COMP3322 Modern Technologies on World Wide Web**

**Lab 4: jQuery**

Overview

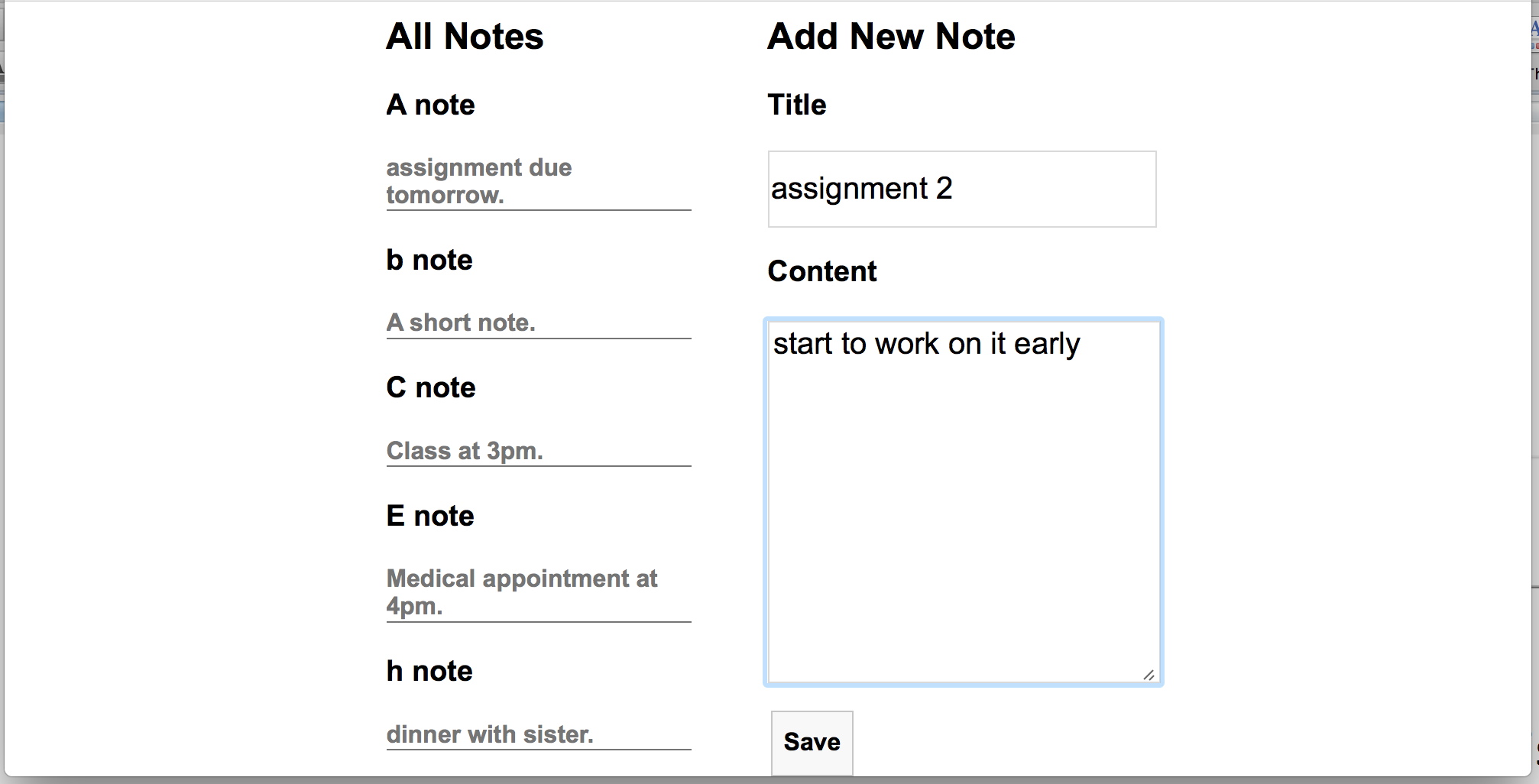
In this lab exercise, we will develop a dynamic webpage to display and update simple notes, using jQuery and PHP. The page shows up as in Fig. 1 below upon loading.

**Fig. 1 upon page load**

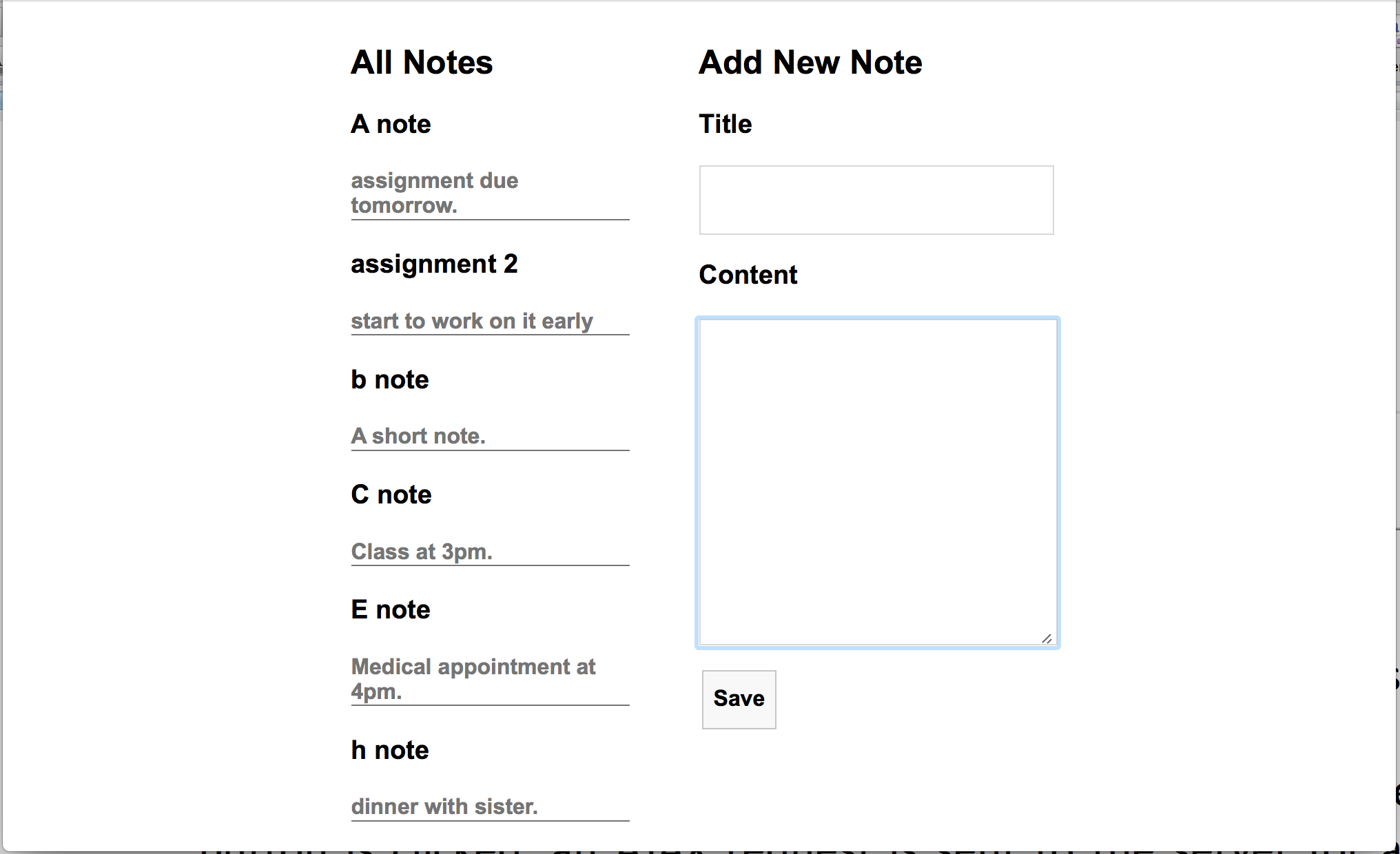


Upon page load, all the notes stored in the database are displayed on the left, with their titles and contents. The titles of the notes should be sorted in ascending alphabetical order. On the right of the page, there are a heading “Add New Note”, two text input fields for creating a new note, and a “Save” button. When the “Save” button is clicked, an AJAX request is sent to the server for adding the new node into the database. For example, suppose you add a new note as shown in Fig. 2. After clicking “Save”, the page display becomes one in Fig. 3. The new node is inserted to the list on the left, according to alphabetic order of its title within titles of all notes, and the text input fields on the right are emptied.

**Fig. 2 after entering title and content of a new note**

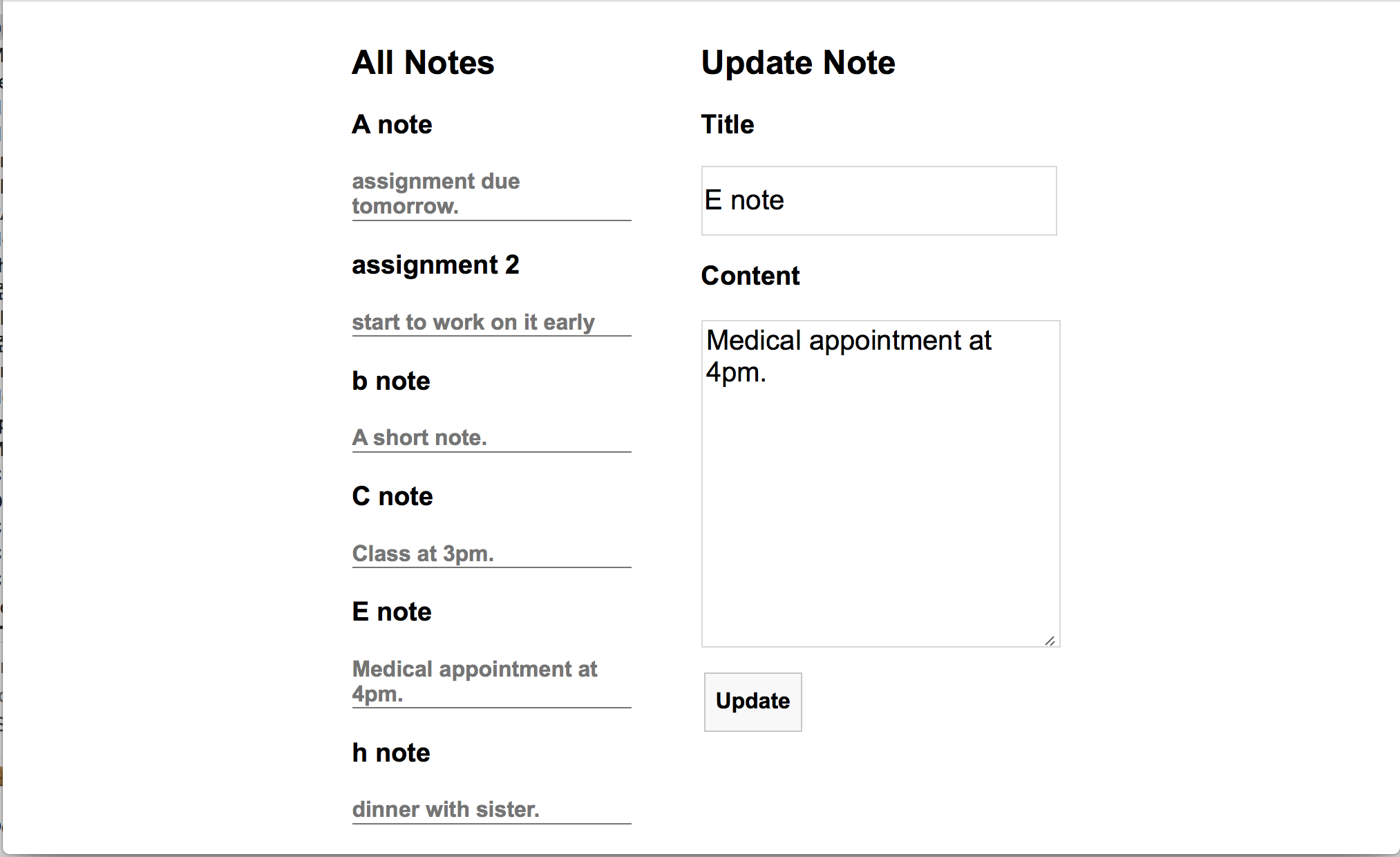


**Fig. 3 after clicking “Save”**



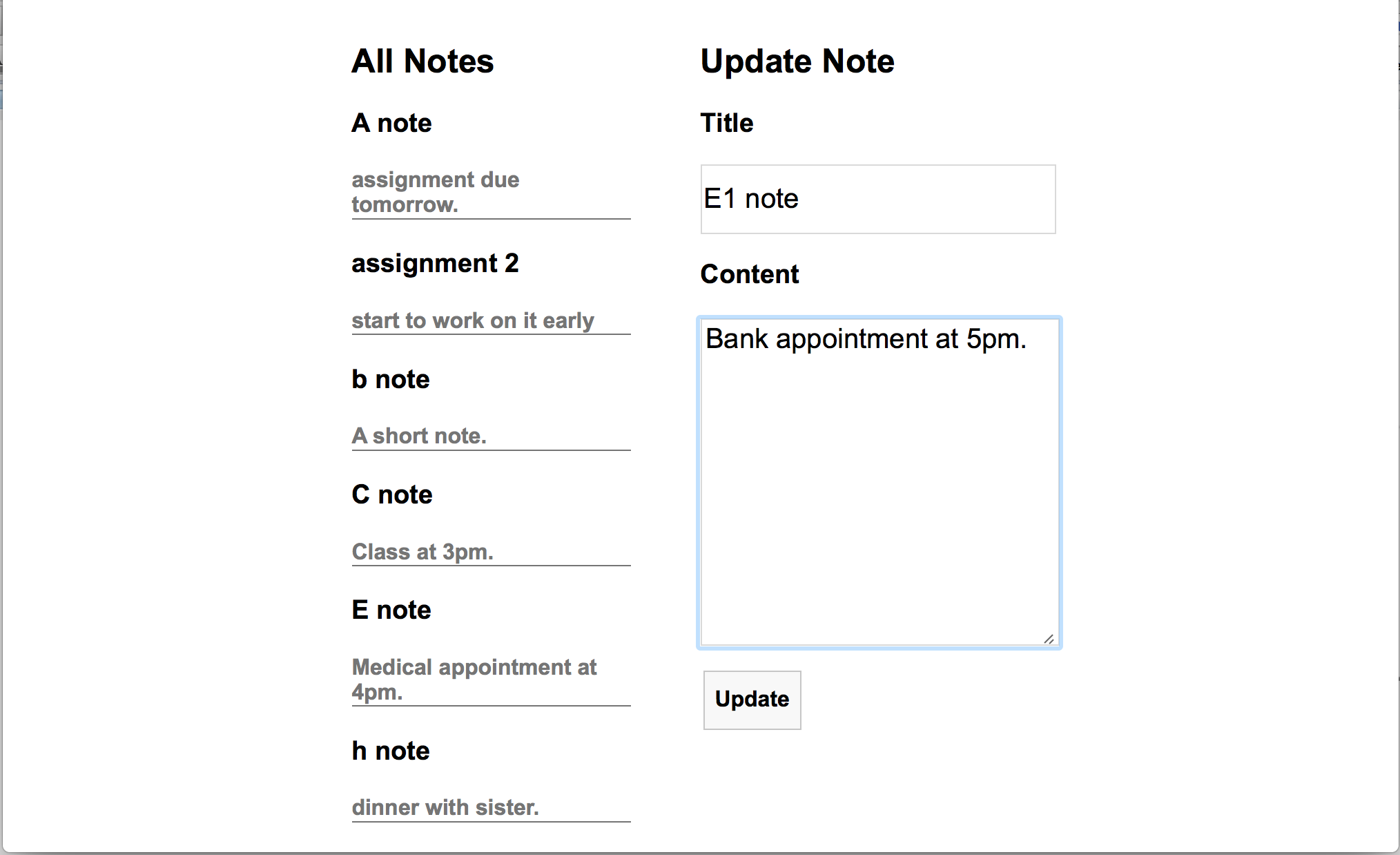
In addition, when you click on the title of a note, the page display changes to one in Fig. 4. On the right of the page, the heading changes to “Update Note”, the title and content of the respective note are displayed in the text fields and can be edited, and the button becomes a “Update” button. When the “Update” button is clicked, an AJAX request is sent to the server for updating the note record in the database.

**Fig. 4 after clicking the note title “E note”**

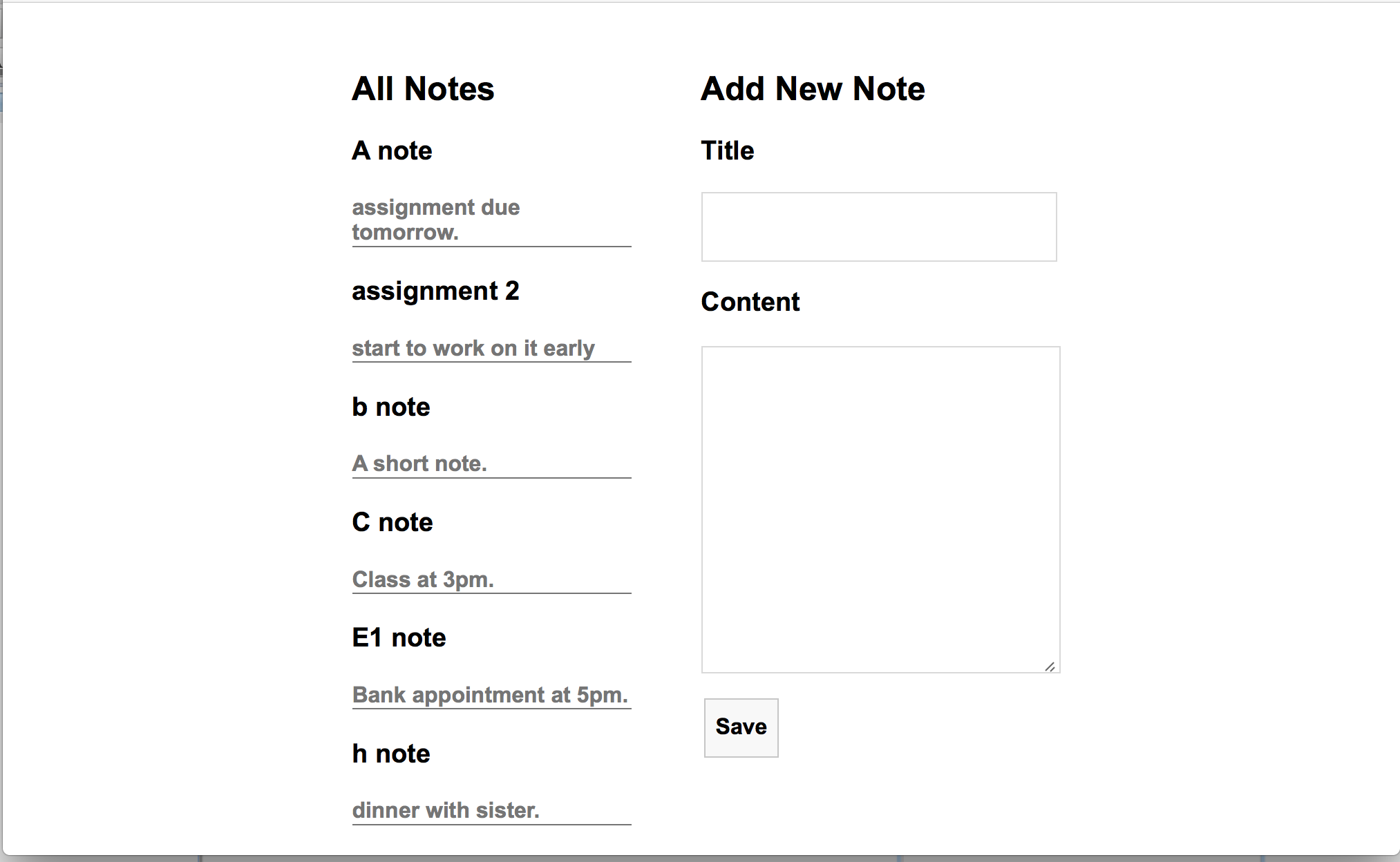


Suppose you update this note as in Fig. 5. After clicking “Update”, the page changes to one in Fig. 6, where the edited note replaces the original note on the left, and the right-hand side of the page shows “Add New Note” and the “Save” button again, and the text input fields are emptied.

**Fig. 5 Editing “E note”**



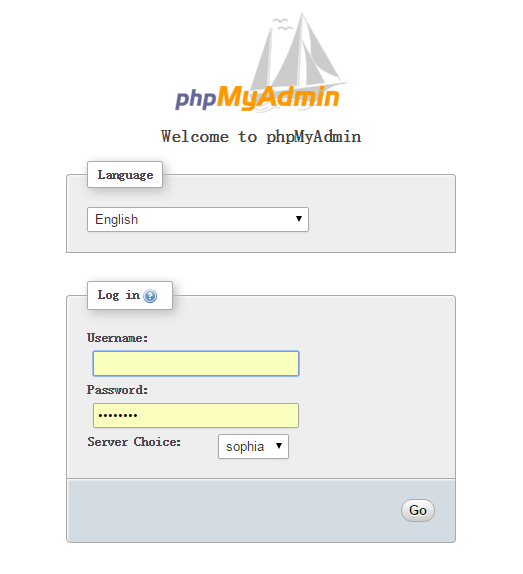
**Fig. 6 After clicking “Update”**



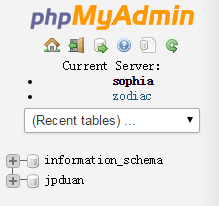
Lab Exercise 1: Set up the Database

The first thing that we do is to set up the database for storing the notes.

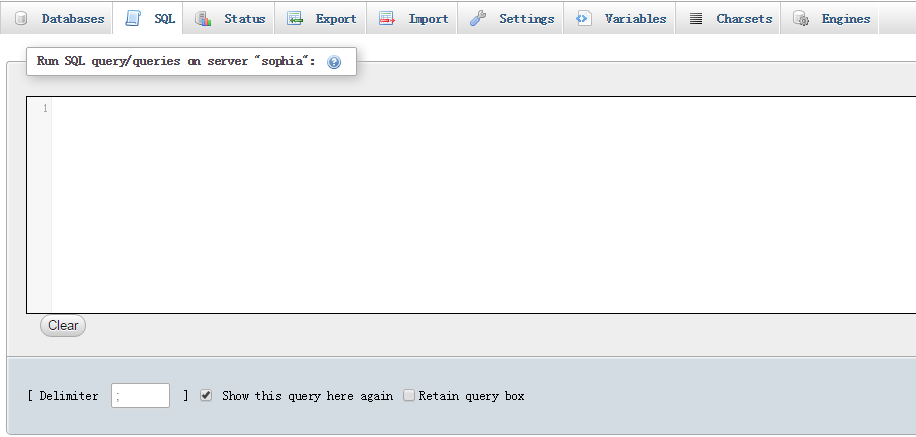
**[Step 1]** Go to https://i.cs.hku.hk/phpmyadmin/. Log in using your MySQL account. Choose the server **Sophia**. We will use the MySQL database hosted on sophia.cs.hku.hk.



**[Step 2]** After logging in, select your database on the left column. Your database name should be the same as your account name.



**[Step 3]** Click the SQL tab and you will see an input field for executing SQL code.



**[Step 4]** Create **notes** table by executing the following SQL code. The **notes** table contains the fields of **id**, **title** and **content**. (All SQL statements are in sql.txt in **lab4\_materials.zip**, which you can copy and paste.) Please insert at least 4-5 notes in the table.

CREATE TABLE notes (

id int NOT NULL,

title varchar(20) NOT NULL,

content longtext,

PRIMARY KEY (id)

)

INSERT INTO notes( id, title, content )

VALUES (

1, 'Title 1', 'A short content.'

)

**Lab Exercise 2: Implement the Dynamic Webpage**

**[Step 1]** Download **lab4\_materials.zip** from Moodle, and unzip it to a folder “lab4”. All the server-side PHP programs, index.html, styl.css, and the jQuery library file have been provided to you. You only need to complete all the **TODO** items in **script.js**. You should upload all files in the folder (you may omit sql.txt) to the following URL for testing your program: [**http://i.cs.hku.hk/~[YourCSID]/lab4/**](http://i.cs.hku.hk/~%5bYourCSID%5d/lab4/). The web page is accessed by retrieving [**http://i.cs.hku.hk/~[YourCSID]/lab4/index.html**](http://i.cs.hku.hk/~%5bYourCSID%5d/lab3/index.php).

**[Step 2]**  Read carefully all the PHP files and index.html provided, and the skeleton implementation in **script.js**, to understand the program workflow. Please change the “username”, “pwd” and “dbname” in all the 3 PHP files to your own CS account name, CS password and CS database name.

**[Step 3]** In **script.js,** when the web page is loaded for the first time, it calls **loadAllNotes()** to retrieve and display all the existing notes from the notes table in the database, set up “click” event handlers for the button elements given in index.html, and display or hide the respective buttons. In **loadAllNotes(),** we use $.getJSON to retrieve all notes from the server side in a JSON string, sort the notes according to alphabetic order of the titles, and display each note as a list item in the notes list. Note that in this lab exercise, when we order the titles in alphabetic order, we can always convert the titles to uppercase first before ordering them; for the “sort” function used in the given code, refer to <http://www.w3schools.com/jsref/jsref_sort.asp>.

Complete TODO task 1-4 in **script.js**.

**[Step 4]** When the user has entered the title and content of a new note and clicked the “Save” button under “Add New Note” (Fig. 2), **saveNewNote()** function is called. The **saveNewNote()** function contacts **handleNewNote.php** for inserting the new note into the database, and then inserts the new note on the web page under “All Notes” (Fig. 3). Note that the code for inserting the new note into the notes list on the webpage has been given; you should understand the code.

Complete TODO tasks 5-9 marked in **saveNewNote()** in **script.js**.

**[Step 5]** When the user clicks the title of a note under “All Notes”, **showUpdate()** function is invoked. In **showUpdate()**, retrieve the title and content of the note being clicked and display them in the input fields under the “Update Note” (Fig. 4). Display the correct header and button accordingly.

Complete TODO tasks 10-13 marked in **showUpdate ()** function in **script.js**.

**[Step 6]** When the “**Update”** button is clicked, the **updateNote()** function is called. It updates the title and content of the note in the database by requesting handleUpdate.php, and updates the note (title and content) in the notes list under “All Notes” on the page (Fig. 6). Note that if the title of the note has been changed, the position of the node in the list may need to be changed, such that all notes are still listed in alphabetic order of their titles. The code for inserting the updated note into the correct position in the notes list on the webpage has been given; you should understand the code.

Complete TODO tasks 14-19 marked in **updateNote()** function in **script.js**.

Submission

Please finish this lab exercise before 23:59 Monday Nov. 7. Upload the following files to i.cs.hku.hk web server under **public\_html/lab4**:

**Index.html, script.js, handleLoadAllNotes.php, handleNewNote.php, handleUpdate.php, jquery-3.1.1.min.js, style.css**

**Please make sure** [**http://i.cs.hku.hk/~[YourCSID]/lab4/index.html**](http://i.cs.hku.hk/~%5bYourCSID%5d/lab3/index.php) **is accessible.**

**You will get the “1” mark for the lab if you fulfill all requirements using jQuery.**