# Final Group Project

Submit Assignment

**Due** Tuesday by 11:59pm **Points** 80 **Submitting** a text entry box **Available** Apr 16 at 12am - May 9 at 11:59pm 24 days

You will each select a partner and complete the following assignment as a group.

## Register Your Group

1. Go to the groups tab in the People page
2. Pick a group that is not already in use.
3. Register yourself in the group with your teammate.
4. Each group will be 2 people.
5. Instructor will randomly assign people who have not selected a group on Friday Afternoon.

## Implementation details

- Pick a reasonable name for your project, make sure the name is "appropriate"
- Names must not contain spaces or special characters
- One team member will create a git repository
  - Make the 2nd team member a full owner
  - Make the following people developers
    - castroa
    - silwals
    - wangy75
    - yeb2
- Each team member shall clone the project to ceclnx01.
- There are five assignments that you must submit:
  - There is a GROUP assignment which is the actual code and project.
    - You will submit ONCE for this assignment and both team members will be given a shared grade for this submission.
  - Individual team-assessment
    - You will answer surveys about your team during and at the end of the project.

## Overview

Create a single page web application that monitors a person's consumption of a set of items in a diary form. The list of items being monitored and the list of authorized users will both be fixed in a database. Only the diary entries will be dynamically added to the database.

### Application

The application flow shall be as follows:

1. User shall connect to the application/index.html.
   1. ALL html code shall be in this file
   2. all javascript shall be in a separate file.
   3. All css shall also be in a separate file
2. User shall be prompted to authenticate.
   1. User entries have been created in the user table for each class member. All password are the same "test"
   2. Using JSON the users credentials shall be sent to the rest api to obtain a token
3. Once properly authenticated the user shall be shown
   - The items list is obtained from making an api call
   - upon clicking a button, a JSON call shall be made to the api which will update their diary with the item and the date/time.
   - The page will then update its entries via javascript
   - A summary of their Diary
   - The last 20 entries of their diary
   - A well formatted set of buttons allowing them to indicate they consumed one of the tracked items.
4. misc:

1. you must always use prepare statements where appropriate

## Authors page

1. At the bottom of the application there will be an hyperlink which hides the application elements (login form or tables) and shows authors' pictures and a brief bio below each of them.
2. In medium or larger screens, the authors will appear in an horizontal format (from left to right); in smaller screens, authors will appear in vertical format (one above the other).
3. These page will have a button to hide the authors and show the corresponding content (login form or tables).

## REST API

### User authentication

- Given user and password will get a token validating the user. If no user is present or the password does not match will return status will == "FAIL"
- passwords are hashed using the php password_hash function
- url: rest.php/v1/user
- method: post
- json_in:
  - user
  - password
- json_out
  - status: "OK" or "FAIL"
  - msg:
  - token: string
- Test:
  - curl -X 'POST' -d '{"user":"test","password":"test"}' https://ceclnx01.cec.miamioh.edu/~campbest/cse383/finalProject/restFinal.php/v1/user

### Return the set of items we are tracking and their key

- rest.php/v1/items
- method: get
- json_in: none
- json_out:
  - pk
  - item
  - status
  - msg
  - items[]
- test:
  - https://ceclnx01.cec.miamioh.edu/~campbest/cse383/finalProject/restFinal.php/v1/items **(https://ceclnx01.cec.miamioh.edu/~campbest/cse383/finalProject/restFinal.php/v1/items)**
  - curl https://ceclnx01.cec.miamioh.edu/~campbest/cse383/finalProject/restFinal.php/v1/items

### Returns the items of a user given a token

- rest.php/items/token
- Call gets the tracked items for a given user
- limit to last 30 items
- Method: GET
- JSON Response:
  - pk
  - item
  - timestamp
  - status: OK or AUTH_FAIL or FAIL
  - msg: text
  - items[]
- test
  - https://ceclnx01.cec.miamioh.edu/~campbest/cse383/finalProject/restFinal.php/v1/items/b34987e72d3dcaed15c4bc2423694948 **(https://ceclnx01.cec.miamioh.edu/~campbest/cse383/finalProject/restFinal.php/v1/items/1db4342013a7c7793edd72c249893a6a095bca71)**

### Return the Item summary

- rest.php/v1/itemsSummary/token
- method: GET
- json_in: none
- json_out
  - item
  - count
  - status
  - msg
  - items[]
- test
  - https://ceclnx01.cec.miamioh.edu/~campbest/cse383/finalProject/restFinal.php/v1/itemsSummary/b34987e72d3dcaed15c4bc2423694948 **(https://ceclnx01.cec.miamioh.edu/~campbest/cse383/finalProject/restFinal.php/v1/itemsSummary/1db4342013a7c7793edd72c249893a6a095bca71)**

### Update the consumed items

- rest.php/v1/items
- Updates item as being consumed.
- method: post
- JSON IN
  - token: string token
  - ItemFK: <key>
- JSON OUT
  - status: OK or AUTH_FAIL or FAIL
  - msg: text
- test (example)
  - curl -X 'POST' -d '{"token":"1db4342013a7c7793edd72c249893a6a095bca71","itemFK":2}' https://ceclnx01.cec.miamioh.edu/~campbest/cse383/finalProject/restFinal.php/v1/items

### Summary of the task

- Get Token
- Get list of items
- Get Items user consumed
- Get Summary of Items
- Update Items Consumed

## Database

- **users**: user table
  - pk
  - user
  - password
  - timestamp
- **diary**: Item Entries
  - pk
  - userFK -> foreign Key to user - not the user but the pk of the user
  - itemFK -> foreign Key to item. Not the item but the PK of the item
  - timestamp
- **diaryItems**: list of items
  - pk: int
  - item: tinytext
- **tokens**
  - pk
  - use - actual user string
  - token - token string created randomly
  - timestamp

You must use a php based data model file separate from your rest code.

# Notes

## SQL Statements

Two more advanced SQL statements are needed in the solution:

- 1st: Get List of items for user
  - to get a list of items for a user, use a join to combine the diary table and the diaryItems table.

```
select diaryItems.item,timestamp from diaryItems left join diary on diaryItems.pk=diary.itemFK  where userFK=? order by timestamp desc
```

This statement combines the two tables, doing the "join" on diaryItems.pk = diary.item.  Then it gets the name of the item and the timestamp in one call instead of having to do two calls.

2nd:

- To get the summary of items, use a "group by"

```
select diaryItems.item,count(timestamp) as count from diaryItems left join diary on diaryItems.pk=diary.itemFK  where userFK=? group by diaryIt
ems.item
```

This provides a list of items and the count of each item.

## Button "trick"

In the buttons you can try the following code to create them:

```
<button class="itemButton" pk=4 onclick='recordItem(this)'>Milk</button>
```

You can create a similar button code like that above for each button, replacing the pk and name for each button.

Then in your code that handles the onclick event, look up the PK so to make the corresponding ajax call.

```
function record(whichButton) {
  var itemPK = $(whichButton).attr('pk');
```

$(whichButton) will hold the button that is clicked.

$(whichButton).attr('pk') will get the value of the attriube called pk for the current button.

Now we know which button was clicked and can send this to the rest-server in our ajax code.

* The buttons are created dynamically from the list of items retrieved from the rest server.

## Tests

- The following tests should all pass
- Copy this to a file on ceclnx01 - http://ceclnx01.cec.miamioh.edu/~campbest/cse383/finalProject/final-tests.txt **(http://ceclnx01.cec.miamioh.edu/~campbest/cse383/finalProject/final-tests.txt)**
- Edit the field UNIQUEID
- Make it executable (chmod a+x final-tests.txt)
- run it "./final-tests.txt"
- Look at each line - tells you what output should look like