

Name: \_\_\_\_\_ Score: \_\_\_\_\_, Date: \_\_\_\_\_

- **100 points + 50 bonus points. Due date: 11:59 pm, 4/20/2019. Late penalty: 10 points off per week after due date.**
- **No resubmission after receiving the score. You MUST completely test the functions before you submit the project.**
- This is an individual project with 2 phases, and this is the 2nd phase.
- You will design and implement an online banking system using MySQL, HTML and PHP.
- You can refer to demo: <http://imc.kean.edu/CPS3740>
- Your project URL must be: <http://eve.kean.edu/~xxxx/CPS3740> where xxxx is your eve ID.
- You must use and implement the table, and store data on the required databases at **imc.kean.edu**.
- Submit your project #1 through the class grade/submission system. <http://imc.kean.edu/students>
- **All the links (p2, search, add, update) should call programs under current folder. No link to eve or other servers.**
- **Note: The program name must be exactly the same as indicated “\*”. Otherwise, your project won’t be graded.**
- **Note: Your project will not be graded if the system detects plagiarism in your programs.**

**Phase #2 will require you to develop the following functions** after the customer logs in successfully. All the functions should be available only after a customer successfully login. Copy your **login1.php** to **login2.php**, and update **login2.php** based on the requirements. Please refer to Figure 1 for the project 2 login welcome main page.

1. \_\_\_\_\_ (5 pts) Your **login2.php** program should set cookie using the **customer id** as the cookie value. If the customer doesn’t successfully login, your program should not set the cookie.
2. \_\_\_\_\_ (5 pts) Your **login2.php** should display all the transactions and total balance in your **Money\_xxxx**.
3. \_\_\_\_\_ (5 pts) On the main page, Add a **logout** link that will call a program to clean/expire the cookie.
4. \_\_\_\_\_ (5 pts) Add **additional** 3 functions - “**Search transaction**” a HTML form with a textbox and a button, “**Add transaction**” a HTML form with a button, “**Display and update transaction**” – a link.
5. \_\_\_\_\_ (5 pts) All the 3 functions (search, add and update) should check if the cookie was set first. If the cookie is not set, your programs should ask users to login first and exit without connecting to the database.
6. \_\_\_\_\_ (5 pts) All the CGI programs should use the POST method, except the **Search** should use GET method
7. **Search** - add search function on the **project 2 home** (shown in Figure 1) which will call a program named “**search.php**” to do the following search functions after clicking on the “Search” button. Your program will use the keywords to do **pattern match** on the **Note** columns for **the logged in customer** in your **Money\_xxxx** table.
  - a. \_\_\_\_\_ (2 pts) If the keyword is empty, an error message “no keyword entered” should be displayed and the program should not connect/access the database.
  - b. \_\_\_\_\_ (3 pts) If the keyword is \*, all the records (rows and cols) should be displayed.
  - c. \_\_\_\_\_ (5 pts) If no record matched, please display “No record found with the search keyword: ....”. No column headers should be displayed.
  - d. \_\_\_\_\_ (5 pts) If records found, please display all matched records in HTML <TABLE> tag format with headers.
  - e. \_\_\_\_\_ (5 pts) Your result should display the **customer name, source name**, NOT the id as shown in Figure 3.

User logout  
Your IP: 100.100.100.100  
You are NOT from Kean University.  
Welcome Customer: Austin Huang  
age: 38  
Address: 1000 Morris Ave., Union, 07083

The transactions for customer Austin Huang are:

ID	Code	Operation	Amount	Date Time	Note
1	xy001	Deposit	4000	2018-12-25 17:10:33	Manually inserted
2	xy002	Withdraw	-2000	2018-12-25 17:10:33	Manually inserted
3	xy003	Deposit	3000	2018-12-25 17:10:33	Manually inserted
4	xy004	Withdraw	-1000	2018-12-25 17:10:33	Manually inserted

Total balance: 4000

Add transaction      [Display and update transaction](#)

Keyword:       Search transaction

Figure 1. New project 2 page

User logout

**Add Transaction**  
Austin Huang current balance is 4000.  
Transaction code:   
☐ Deposit   ☐ Withdraw  
 Amount:   
 Select a Source:  ATM  
 Note:

Figure 2. Add transaction page

The transactions in customer New New records matched keyword \* are:

ID	Code	Type	Amount	Date Time	Note	Source
87	sadsad	Withdraw	-2	2019-03-11 20:59:08	sdsdk	Branch
90	182	Deposit	1000	2019-03-12 00:00:58	online	Online
91	183	Deposit	1000	2019-03-12 00:15:34	online	Online

Total balance: 1998

Figure 3. The search result display

8. If customers click on “**Add transaction**” button, a program named “**add\_transaction.php**” should be called to do the following functions. Refer to Figure 2 for the graphic user interface.
  - a. \_\_\_\_\_ (5 pts) **login2.php** should pass **customer name** to **add\_transaction.php**. The project 2 main page should show the **customer name** and the **balance** at the top.
  - b. \_\_\_\_\_ (5 pts) The main page should allow the customer to enter transaction **code** in **textbox**, select **deposit** or **withdraw** (no default) from **two radio buttons**, enter the **amount** in **textbox**, select **source name** in a **dropdown** list (no default), and a **submit button** to add the data into your **Money\_xxxx** table.
  - c. \_\_\_\_\_ (10 pts) Your program should retrieve all the **source id** and **names** first from **CPS3740.Sources** table, and only display the **source names** in the dropdown list for the customer to select, but you need to pass the **source id** to the next program **insert\_transaction.php**.
9. After customer clicks the **Submit** button, a program named “**insert\_transaction.php**” should be called to insert the data into your **Money\_xxxx** table after verifying the input based on the following requirements.
  - a. \_\_\_\_\_ (5 pts) Show an error message indicating the problem if **the amount is empty, amount <=0, the type (deposit, withdraw) or a source is not selected**.
  - b. \_\_\_\_\_ (10 pts) Show an error message indicating the problem if **the balance is less than the withdraw amount**. You have to retrieve and check **the latest balance** for **the logged in customer** in your **Money\_xxxx** table before inserting the withdraw transaction because the balance in the program might be outdated.
  - c. \_\_\_\_\_ (5 pts) Show an error message indicating the problem if the **Same Code exists** in the table.
  - d. \_\_\_\_\_ (5 pts) If there is no error, insert the data into your **Money\_xxxx** table. Please note that you need to retrieve **customer id** from the **browser cookie** and insert the **customer id** into the field **cid**. You also need to insert a **negative amount** if it is **withdraw**.
  - e. \_\_\_\_\_ (5 pts) If there is a data error, **no data should be added into your Money\_xxxx table**.
13. (bonus) If customer clicks on “**Display and update transaction**” link, a program named “**display\_update\_transcation.php**” should show all the customer’s transactions in your **Money\_xxxx** table on a web page as shown in Figure 4.
  - a. \_\_\_\_\_ (10 pts) The **Note** field is **updateable** and you should **highlight it in yellow**. All other columns are **Non-updateable fields**.
  - b. \_\_\_\_\_ (5 pts) Add a **delete** column that has the HTML **checkbox** for each record after the **Note** column.
14. (bonus) If customer clicks on “**Update transaction**” button, a program named “**update\_transcation.php**” should be called to do the **followings**:
  - a. \_\_\_\_\_ (5 pts) if a record’s **delete** checkbox is checked, that record should be deleted from your **Money\_xxxx** table. A message “The Code xxxx has been deleted from the database.” should be displayed for **each deleted record**.
  - b. \_\_\_\_\_ (5 pts) if a record’s delete checkbox is not checked, but the **Note** was updated, you should update the **Note** column of the **corresponding record** in your **Money\_xxxx** table. A message “The Note for code xxxx has been updated in the database.” should be displayed for each updated record.
  - c. \_\_\_\_\_ (10 pts) Your program should be able to handle mixing delete and update operations. All the change and delete should be successfully done by **ONE** click, and other not changed records should not be updated.
  - d. \_\_\_\_\_ (10 pts) Display a message indicating **the # of deleted records and the # of updated records**.
  - e. \_\_\_\_\_ (5 pts) Display the corresponding error message and indicate the **Code** of the record if an error happens during the delete or update.

[User logout](#)  
You can only update **Note** column.

ID	Code	Amount	Operation	Source	Date Time	Note	Delete
1	xy001	4000	D	ATM	2018-12-26 02:12:01	Manually inserted	<input type="checkbox"/>
2	xy002	-2000	W	Online	2018-12-26 02:12:01	Manually inserted	<input type="checkbox"/>
3	xy003	3000	D	Online	2018-12-26 02:12:01	Manually inserted	<input type="checkbox"/>
4	xy004	-1000	W	Brarnch	2018-12-26 02:12:01	Manually inserted	<input type="checkbox"/>

Total balance: 4000  
Update transaction

**Figure 4.** Updateable fields are shown in yellow color, and non-updateable fields white color.