Jordan Lian Engineering Database Systems – IE 3425

Final Project – Northeastern Goods December 10, 2018

Report for Northeastern Goods Database based off HTML, Python, and MYSQL

Home Page URL: http://www1.coe.neu.edu/~jlian/Final%20Project/Central%20Web%20Page.html

DB Backend)

1. Table schema of DB tables in MYSQL: screenshots of tables

CUSTOMER Table

```
mysql> DESCRIBE CUSTOMER;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| CUS_ID     | int(2)      | NO   | PRI | NULL    |       |
| CUS_FNAME  | varchar(20) | YES  |     | NULL    |       |
| CUS_LNAME  | varchar(20) | YES  |     | NULL    |       |
| CUS_GENDER | varchar(2)  | YES  |     | NULL    |       |
| CUS_DOB    | date        | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

EMPLOYEE Table

```
mysql> DESCRIBE EMPLOYEE;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| EMP_ID     | int(2)      | NO   | PRI | NULL    |       |
| EMP_FNAME  | varchar(20) | YES  |     | NULL    |       |
| EMP_LNAME  | varchar(20) | YES  |     | NULL    |       |
| EMP_DOB    | date        | YES  |     | NULL    |       |
| EMP_HIRE   | date        | YES  |     | NULL    |       |
| EMP_GENDER | varchar(2)  | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

EMPLOYEE SCHOOL Table

```
mysql> DESCRIBE EMPLOYEE_SCHOOL;
+--------+------------+------+-----+---------+-------+
| Field  | Type       | Null | Key | Default | Extra |
+--------+------------+------+-----+---------+-------+
| EMP_ID | int(2)     | YES  | MUL | NULL    |       |
| SCH_ID | varchar(4) | YES  | MUL | NULL    |       |
+--------+------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

NORTHEASTERN SCHOOL Table

```
mysql> DESCRIBE NORTHEASTERN_SCHOOL;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| SCH_ID   | varchar(4)  | NO   | PRI | NULL    |       |
| SCH_NAME | varchar(50) | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

MANAGER Table

```
mysql> DESCRIBE MANAGER;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| MAN_ID     | int(2)      | NO   | PRI | NULL    |       |
| MAN_FNAME  | varchar(20) | YES  |     | NULL    |       |
| MAN_LNAME  | varchar(20) | YES  |     | NULL    |       |
| MAN_DOB    | date        | YES  |     | NULL    |       |
| MAN_HIRE   | date        | YES  |     | NULL    |       |
| MAN_GENDER | varchar(2)  | YES  |     | NULL    |       |
| EMP_ID     | int(2)      | YES  | MUL | NULL    |       |
+------------+-------------+------+-----+---------+-------+
7 rows in set (0.00 sec)
```

LINE Table

```
mysql> DESCRIBE LINE;
+--------------+--------------+------+-----+---------+-------+
| Field        | Type         | Null | Key | Default | Extra |
+--------------+--------------+------+-----+---------+-------+
| ORD_NUM      | int(2)       | NO   | PRI | NULL    |       |
| ORD_DATE     | date         | YES  |     | NULL    |       |
| PROD_CODE    | int(2)       | YES  | MUL | NULL    |       |
| PROD_PRICE   | decimal(6,2) | YES  |     | NULL    |       |
| ORD_UNITS    | int(2)       | YES  |     | NULL    |       |
| ORD_SUBTOTAL | decimal(6,2) | YES  |     | NULL    |       |
| ORD_TAX      | decimal(6,2) | YES  |     | NULL    |       |
| ORD_TOT      | decimal(6,2) | YES  |     | NULL    |       |
+--------------+--------------+------+-----+---------+-------+
8 rows in set (0.01 sec)
```

ORDERS Table

```
mysql> DESCRIBE ORDERS;
+---------+--------+------+-----+---------+-------+
| Field   | Type   | Null | Key | Default | Extra |
+---------+--------+------+-----+---------+-------+
| ORD_NUM | int(2) | YES  | MUL | NULL    |       |
| CUS_ID  | int(2) | YES  | MUL | NULL    |       |
+---------+--------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

PRODUCT Table

```
mysql> DESCRIBE PRODUCT;
+------------+--------------+------+-----+---------+-------+
| Field      | Type         | Null | Key | Default | Extra |
+------------+--------------+------+-----+---------+-------+
| PROD_CODE  | int(2)       | NO   | PRI | NULL    |       |
| PROD_NAME  | varchar(30)  | YES  |     | NULL    |       |
| PROD_PRICE | decimal(6,2) | YES  |     | NULL    |       |
+------------+--------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

2. Table data of DB tables in MYSQL: screenshots of MYSQL tables showing rows

CUSTOMER Table

```
mysql> SELECT * FROM CUSTOMER;
+--------+-----------+-----------+------------+------------+
| CUS_ID | CUS_FNAME | CUS_LNAME | CUS_GENDER | CUS_DOB    |
+--------+-----------+-----------+------------+------------+
|      1 | Elena     | Shelton   | F          | 1999-02-01 |
|      2 | Reese     | Lindsey   | M          | 1999-06-20 |
|      3 | Heather   | Klein     | F          | 1998-08-07 |
|      4 | Neil      | McCann    | M          | 1999-04-21 |
|      5 | Josh      | Hooper    | M          | 1997-05-02 |
|      6 | Demarcus  | Moody     | M          | 2000-10-31 |
|      7 | Ayaan     | Galvan    | F          | 1996-07-25 |
|      8 | Ishaan    | Oneill    | M          | 1999-11-05 |
|      9 | Araceli   | Floyd     | F          | 1999-01-18 |
|     10 | Carter    | Weeks     | M          | 2000-03-06 |
+--------+-----------+-----------+------------+------------+
10 rows in set (0.01 sec)
```

EMPLOYEE Table

```
mysql> SELECT * FROM EMPLOYEE;
+--------+-------------+-----------+------------+------------+------------+
| EMP_ID | EMP_FNAME   | EMP_LNAME | EMP_DOB    | EMP_HIRE   | EMP_GENDER |
+--------+-------------+-----------+------------+------------+------------+
|      1 | Thomas      | Minutella | 1998-06-22 | 2018-01-02 | M          |
|      2 | Mason       | Kufs      | 1999-06-17 | 2018-01-02 | M          |
|      3 | Jason       | Harris    | 1997-12-11 | 2018-01-10 | M          |
|      4 | Charlie     | Peters    | 1996-09-27 | 2018-02-16 | M          |
|      5 | Max-Antoine | Bonnet    | 1999-01-27 | 2018-02-27 | M          |
|      6 | Matthew     | Thalmann  | 1998-06-28 | 2018-03-01 | M          |
|      7 | Max         | Corkran   | 1999-01-11 | 2018-03-15 | M          |
|      8 | Kelly       | Marchese  | 1999-07-28 | 2018-03-23 | F          |
|      9 | Caleb       | Driesman  | 1999-03-14 | 2018-04-16 | M          |
|     10 | Justin      | Adams     | 1997-11-19 | 2018-04-20 | M          |
|     11 | Jordan      | Lian      | 1999-06-19 | 2018-01-01 | M          |
|     12 | Abe         | Zeid      | 1990-12-25 | 2018-01-02 | M          |
|     13 | Bob         | Ross      | 1942-10-19 | 2018-01-03 | M          |
+--------+-------------+-----------+------------+------------+------------+
13 rows in set (0.01 sec)
```

EMPLOYEE SCHOOL Table

```
mysql> SELECT * FROM EMPLOYEE_SCHOOL;
+--------+--------+
| EMP_ID | SCH_ID |
+--------+--------+
|      1 | COS    |
|      2 | BCHS   |
|      3 | CAMD   |
|      4 | COE    |
|      5 | CPS    |
|      6 | DMB    |
|      7 | SL     |
|      8 | CCIS   |
|      9 | CSSH   |
|     10 | COE    |
+--------+--------+
10 rows in set (0.00 sec)
```

NORTHEASTERN SCHOOL Table

```
mysql> SELECT * FROM NORTHEASTERN_SCHOOL;
+--------+-----------------------------------------------+
| SCH_ID | SCH_NAME                                      |
+--------+-----------------------------------------------+
| BCHS   | Bouvé College of Health Sciences              |
| CAMD   | College of Arts                               |
| CCIS   | College of Computer and Information Science   |
| COE    | College of Engineering                        |
| COS    | College of Science                            |
| CPS    | College of Professional Studies               |
| CSSH   | College of Social Sciences and Humanities     |
| DMB    | D'Amore-McKim School of Business              |
| SL     | School of Law                                 |
+--------+-----------------------------------------------+
9 rows in set (0.00 sec)
```

MANAGER Table

```
mysql> SELECT * FROM MANAGER;
+--------+-----------+-----------+------------+------------+------------+--------+
| MAN_ID | MAN_FNAME | MAN_LNAME | MAN_DOB    | MAN_HIRE   | MAN_GENDER | EMP_ID |
+--------+-----------+-----------+------------+------------+------------+--------+
|      1 | Jordan    | Lian      | 1999-06-19 | 2018-01-01 | M          |     11 |
|      2 | Abe       | Zeid      | 1990-12-25 | 2018-01-02 | M          |     12 |
|      3 | Bob       | Ross      | 1942-10-19 | 2018-01-03 | M          |     13 |
+--------+-----------+-----------+------------+------------+------------+--------+
3 rows in set (0.01 sec)
```

LINE Table

```
mysql> SELECT * FROM LINE;
+---------+------------+-----------+------------+-----------+--------------+---------+---------+
| ORD_NUM | ORD_DATE   | PROD_CODE | PROD_PRICE | ORD_UNITS | ORD_SUBTOTAL | ORD_TAX | ORD_TOT |
+---------+------------+-----------+------------+-----------+--------------+---------+---------+
|       1 | 2018-01-04 |         2 |       1.00 |         2 |         2.00 |    0.13 |    2.13 |
|       2 | 2018-01-10 |         6 |       2.50 |         4 |        10.00 |    0.63 |   10.63 |
|       3 | 2018-02-01 |         6 |       2.50 |         1 |         2.50 |    0.16 |    2.66 |
|       4 | 2018-02-20 |         1 |       1.75 |         6 |        10.50 |    0.66 |   11.16 |
|       5 | 2018-02-27 |        10 |       1.00 |         1 |         1.00 |    0.06 |    1.06 |
|       6 | 2018-03-05 |         9 |       2.00 |         3 |         6.00 |    0.38 |    6.38 |
|       7 | 2018-04-01 |         4 |       3.50 |         5 |        17.50 |    1.09 |   18.59 |
|       8 | 2018-04-20 |         3 |       3.00 |         4 |        12.00 |    0.75 |   12.75 |
|       9 | 2018-05-16 |         7 |       3.25 |         5 |        16.25 |    1.02 |   17.27 |
|      10 | 2018-08-30 |         5 |       2.75 |         1 |         2.75 |    0.17 |    2.92 |
|      11 | 2018-09-02 |         9 |       2.00 |        10 |        20.00 |    1.25 |   21.25 |
|      12 | 2018-09-20 |         8 |       1.75 |         8 |        14.00 |    0.88 |   14.88 |
|      13 | 2018-10-12 |         6 |       2.50 |         2 |         5.00 |    0.31 |    5.31 |
|      14 | 2018-10-31 |         1 |       1.75 |         5 |         8.75 |    0.55 |    9.30 |
|      15 | 2018-11-03 |         7 |       3.25 |         6 |        19.50 |    1.22 |   20.72 |
+---------+------------+-----------+------------+-----------+--------------+---------+---------+
15 rows in set (0.00 sec)
```

ORDERS Table

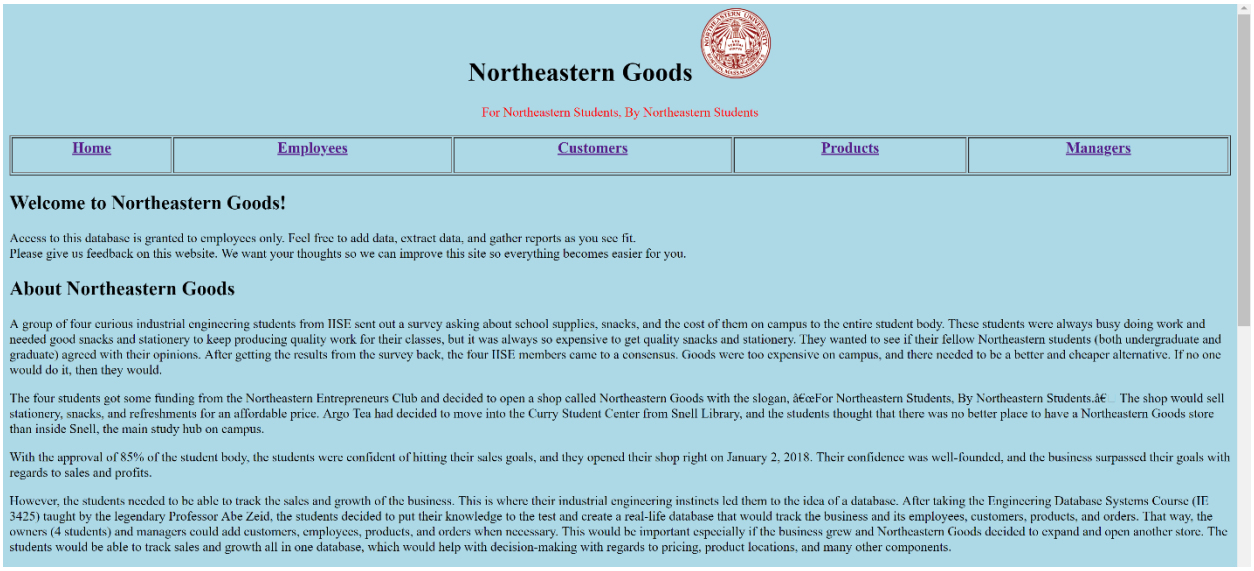
```
mysql> SELECT * FROM ORDERS;
+---------+--------+
| ORD_NUM | CUS_ID |
+---------+--------+
|       1 |      2 |
|       2 |      6 |
|       3 |      6 |
|       4 |      1 |
|       5 |     10 |
|       6 |      9 |
|       7 |      4 |
|       8 |      3 |
|       9 |      7 |
|      10 |      5 |
|      11 |      9 |
|      12 |      8 |
|      13 |      6 |
|      14 |      1 |
|      15 |      7 |
+---------+--------+
15 rows in set (0.00 sec)
```

PRODUCT Table

```
mysql> SELECT * FROM PRODUCT;
+-----------+--------------------+------------+
| PROD_CODE | PROD_NAME          | PROD_PRICE |
+-----------+--------------------+------------+
|         1 | Trail Mix          |       1.75 |
|         2 | Pack of Starburst  |       1.00 |
|         3 | Mechanical Pencils |       3.00 |
|         4 | Ballpoint Pens     |       3.50 |
|         5 | Spiral Notebook    |       2.75 |
|         6 | Looseleaf Paper    |       2.50 |
|         7 | Plastic Folder     |       3.25 |
|         8 | Coffee             |       1.75 |
|         9 | Orange Juice       |       2.00 |
|        10 | Water              |       1.00 |
+-----------+--------------------+------------+
10 rows in set (0.00 sec)
```

Front end Access GUI) (for items 1-7 submit screenshots of the HTML code and browser window)

1. Home webpage



Northeastern Goods

For Northeastern Students, By Northeastern Students

Home	Employees	Customers	Products	Managers

Welcome to Northeastern Goods!

Access to this database is granted to employees only. Feel free to add data, extract data, and gather reports as you see fit.
Please give us feedback on this website. We want your thoughts so we can improve this site so everything becomes easier for you.

About Northeastern Goods

A group of four curious industrial engineering students from IISE sent out a survey asking about school supplies, snacks, and the cost of them on campus to the entire student body. These students were always busy doing work and needed good snacks and stationery to keep producing quality work for their classes, but it was always so expensive to get quality snacks and stationery. They wanted to see if their fellow Northeastern students (both undergraduate and graduate) agreed with their opinions. After getting the results from the survey back, the four IISE members came to a consensus. Goods were too expensive on campus, and there needed to be a better and cheaper alternative. If no one would do it, then they would.

The four students got some funding from the Northeastern Entrepreneurs Club and decided to open a shop called Northeastern Goods with the slogan, â€œFor Northeastern Students, By Northeastern Students.â€ The shop would sell stationery, snacks, and refreshments for an affordable price. Argo Tea had decided to move into the Curry Student Center from Snell Library, and the students thought that there was no better place to have a Northeastern Goods store than inside Snell, the main study hub on campus.

With the approval of 85% of the student body, the students were confident of hitting their sales goals, and they opened their shop right on January 2, 2018. Their confidence was well-founded, and the business surpassed their goals with regards to sales and profits.

However, the students needed to be able to track the sales and growth of the business. This is where their industrial engineering instincts led them to the idea of a database. After taking the Engineering Database Systems Course (IE 3425) taught by the legendary Professor Abe Zeid, the students decided to put their knowledge to the test and create a real-life database that would track the business and its employees, customers, products, and orders. That way, the owners (4 students) and managers could add customers, employees, products, and orders when necessary. This would be important especially if the business grew and Northeastern Goods decided to expand and open another store. The students would be able to track sales and growth all in one database, which would help with decision-making with regards to pricing, product locations, and many other components.

Anonymous Feedback

Name

First Name:
Middle Name:
Last Name:

Select your grade/year at Northeastern

Freshman Sophomore Middler Junior Senior

Select the tabs you visited

Employee Customer Product Manager

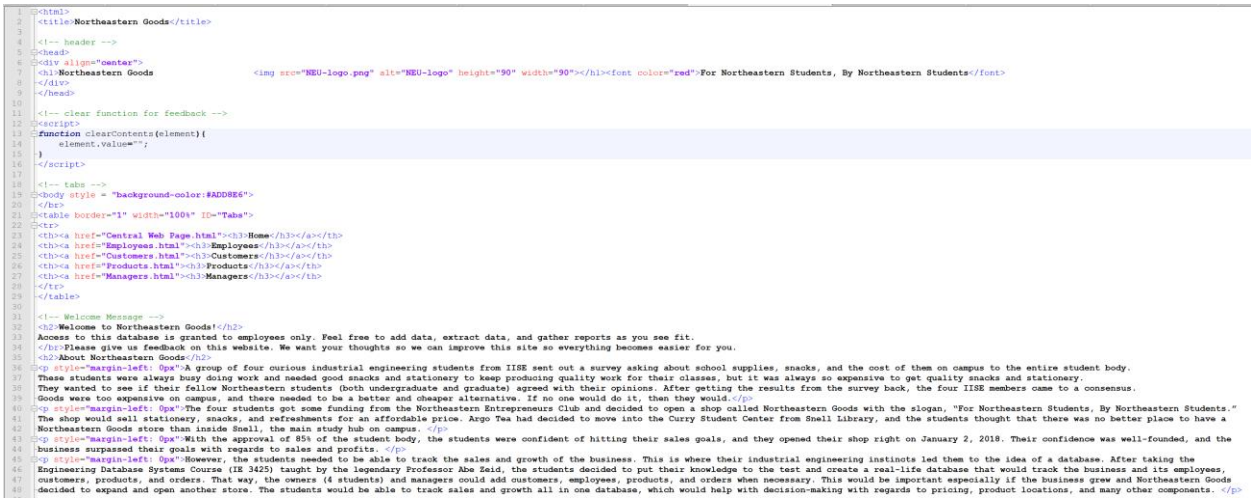
Who is your favorite professor

Abe Zeid
aBe zEiD
ABE ZEID
Definitely Abe Zeid

Comments

We value your feedback.
Please tell us how we can better serve you.

Submit Clear



```
<html>
<title>Northeastern Goods</title>

<!-- header -->
<head>
<div align="center">
<h1>Northeastern Goods                <img src="NEU-logo.png" alt="NEU-logo" height="90" width="90"></h1><font color="red">For Northeastern Students, By Northeastern Students</font>
</div>
</head>

<!-- clear function for feedback -->
<script>
function clearContents(element){
    element.value="";
}
</script>

<!-- tabs -->
<body style = "background-color:#ADD8E6">
</br>
<table border="1" width="100%" ID="Tabs">
<tr>
<th><a href="Central Web Page.html"><h3>Home</h3></a></th>
<th><a href="Employees.html"><h3>Employees</h3></a></th>
<th><a href="Customers.html"><h3>Customers</h3></a></th>
<th><a href="Products.html"><h3>Products</h3></a></th>
<th><a href="Managers.html"><h3>Managers</h3></a></th>
</tr>
</table>

<!-- Welcome Message -->
<h2>Welcome to Northeastern Goods!</h2>
Access to this database is granted to employees only. Feel free to add data, extract data, and gather reports as you see fit.
</br>Please give us feedback on this website. We want your thoughts so we can improve this site so everything becomes easier for you.
<h2>About Northeastern Goods</h2>
<p style="margin-left: 0px">A group of four curious industrial engineering students from IISE sent out a survey asking about school supplies, snacks, and the cost of them on campus to the entire student body.
These students were always busy doing work and needed good snacks and stationery to keep producing quality work for their classes, but it was always so expensive to get quality snacks and stationery.
They wanted to see if their fellow Northeastern students (both undergraduate and graduate) agreed with their opinions. After getting the results from the survey back, the four IISE members came to a consensus.
Goods were too expensive on campus, and there needed to be a better and cheaper alternative. If no one would do it, then they would.</p>
<p style="margin-left: 0px">The four students got some funding from the Northeastern Entrepreneurs Club and decided to open a shop called Northeastern Goods with the slogan, "For Northeastern Students, By Northeastern Students."
The shop would sell stationery, snacks, and refreshments for an affordable price. Argo Tea had decided to move into the Curry Student Center from Snell Library, and the students thought that there was no better place to have a
Northeastern Goods store than inside Snell, the main study hub on campus. </p>
<p style="margin-left: 0px">With the approval of 85% of the student body, the students were confident of hitting their sales goals, and they opened their shop right on January 2, 2018. Their confidence was well-founded, and the
business surpassed their goals with regards to sales and profits. </p>
<p style="margin-left: 0px">However, the students needed to be able to track the sales and growth of the business. This is where their industrial engineering instincts led them to the idea of a database. After taking the
Engineering Database Systems Course (IE 3425) taught by the legendary Professor Abe Zeid, the students decided to put their knowledge to the test and create a real-life database that would track the business and its employees,
customers, products, and orders. That way, the owners (4 students) and managers could add customers, employees, products, and orders when necessary. This would be important especially if the business grew and Northeastern Goods
decided to expand and open another store. The students would be able to track sales and growth all in one database, which would help with decision-making with regards to pricing, product locations, and many other components. </p>
```

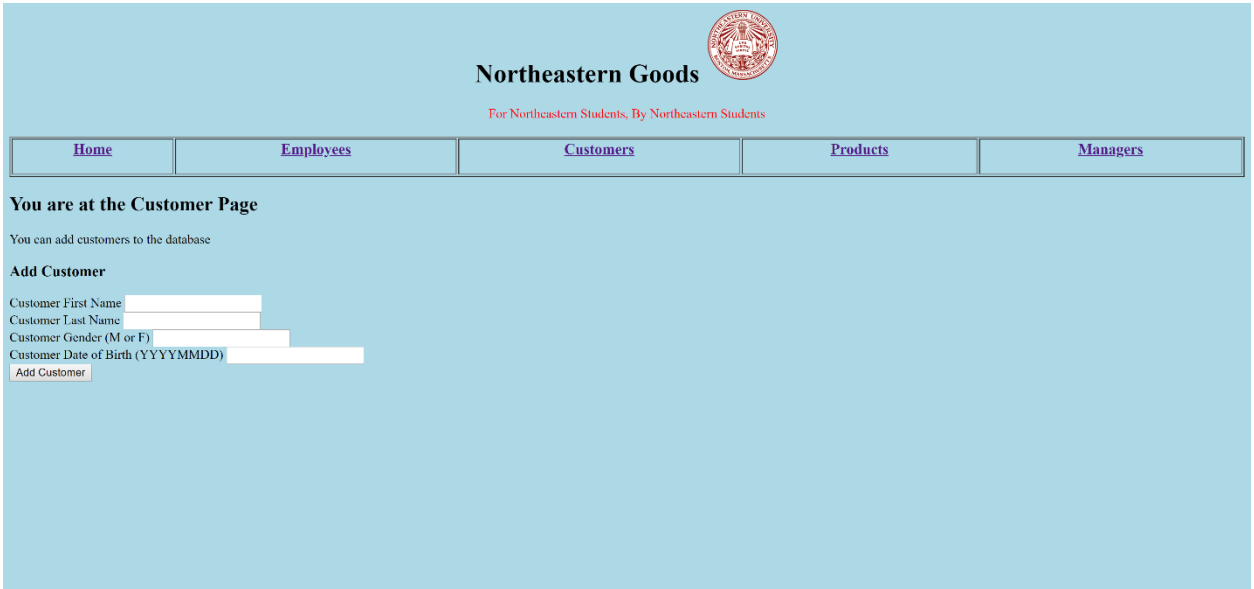
```
<!-- Feedback form -->
<form method = "post" action = "Feedback.psp">

<h2>Anonymous Feedback</h2>

<h3>Name</h3>
First Name: <input type = 'text' name='fName'/></br>
Middle Name: <input type = 'text' name='mName'/></br>
Last Name: <input type = 'text' name='lName'/><br/>

<h3>Select your grade/year at Northeastern</h3>
<input type = "radio" name="year" value = "Freshman"/>Freshman
<input type = "radio" name="year" value = "Sophomore"/>Sophomore
<input type = "radio" name="year" value = "Middler"/>Middler
<input type = "radio" name="year" value = "Junior"/>Junior
<input type = "radio" name="year" value = "Senior"/>Senior</br>

<h3>Select the tabs you visited</h3>
<input type = "checkbox" name="tab" value = "Employee"/>Employee
<input type = "checkbox" name="tab" value = "Customer"/>Customer
<input type = "checkbox" name="tab" value = "Product"/>Product
<input type = "checkbox" name="tab" value = "Manager"/>Manager

<h3>Who is your favorite professor</h3>
<select name = "AbeZeid" multiple>
<option value = "Abe Zeid">Abe Zeid</option>
<option value = "aBe zEiD">aBe zEiD</option>
<option value = "ABE ZEID">ABE ZEID</option>
<option value = "Definitely Abe Zeid">Definitely Abe Zeid</option>
<option value = "Are you kidding me Abe Zeid is THE ONLY CHOICE">Are you kidding me Abe Zeid is THE ONLY CHOICE</option>
<option value = "Professor Abe Zeid is the greatest professor of all time">Professor Abe Zeid is the greatest professor of all time</option>
</select>

<h3>Comments</h3>
<textarea name = "feedback" rows="4" cols="50" onFocus = "clearContents (this)">
We value your feedback.
Please tell us how we can better serve you.
</textarea>
</br>
<input type = 'submit' name = 'Send it' value = 'Submit'/>
<input type = 'reset' name = 'clear it' value = 'Clear'/>
</form>
</body>
</html>
```

2. At least 4 web pages

Employee Web Page

```
<html>
<title>Northeastern Goods</title>

<!-- header -->
<head>
<div align="center">
<h1>Northeastern Goods                <img src="NEU-logo.png" alt="NEU-logo" height="90" width="90"></h1><font color="red">For Northeastern Students, By Northeastern Students</font>
</div>
</head>

<!-- tabs -->
<body style = "background-color:#ADD8E6">
</br>
<table border="1" width="100%" ID="Tabs">
<tr>
<th><a href="Central Web Page.html"><h3>Home</h3></a></th>
<th><a href="Employees.html"><h3>Employees</h3></a></th>
<th><a href="Customers.html"><h3>Customers</h3></a></th>
<th><a href="Products.html"><h3>Products</h3></a></th>
<th><a href="Managers.html"><h3>Managers</h3></a></th>
</tr>
</table>

<!-- Welcome Message -->
<h2>You are at the Employee Page</h2>

<p style="margin-left: 0px">You can do a few things. </p>
<ul>
<li> Add employees to the database </li>
<li> Find the total number of employees </li>
</ul>
<!-- Forms -->

<!-- Add Employee -->
<h3>Add Employee</h3>
<form method = "post" action = "Add Employees.psp">
Employee First Name <input type= "text" name="EMP_FNAME"/><br/>
Employee Last Name <input type = "text" name="EMP_LNAME"/><br/>
Employee Date of Birth (YYYYMMDD) <input type = "text" name="EMP_DOB"/><br/>
Employee Date of Hire (YYYYMMDD) <input type = "text" name="EMP_HIRE"/><br>
Employee Gender (M or F) <input type = "text" name="EMP_GENDER"/><br/>
<input type= "submit" name= "add" value= "Add Employee"/>
</form>

<!-- Employee Report + Total Employees -->
<form method = "post" action = "Total Employees.psp">
<h3>When you click this button you will find how many total employees there are </h3>
<input type = "submit" name="add" value="Total Employees">
</form>
</form>
</body>
</html>
```

Customer Web Page



```
<html>
<title>Northeastern Goods</title>

<!-- header -->
<head>
<div align="center">
<h1>Northeastern Goods                  <img src="NEU-logo.png" alt="NEU-logo" height="90" width="90"></h1><font color="red">For Northeastern Students, By Northeastern Students</font>
</div>
</head>

<!-- tabs -->
<body style = "background-color:#ADD8E6">
</br>
<table border="1" width="100%" ID="Tabs">
<tr>
<th><a href="Central Web Page.html"><h3>Home</h3></a></th>
<th><a href="Employees.html"><h3>Employees</h3></a></th>
<th><a href="Customers.html"><h3>Customers</h3></a></th>
<th><a href="Products.html"><h3>Products</h3></a></th>
<th><a href="Managers.html"><h3>Managers</h3></a></th>
</tr>
</table>

<!-- Welcome Message -->
<h2>You are at the Customer Page</h2>
<p style="margin-left: 0px">You can add customers to the database </p>

<!-- Forms -->

<!-- Add Customer -->
<h3>Add Customer</h3>
<form method = "post" action = "Add Customers.psp">
Customer First Name <input type= "text" name="CUS_FNAME"/><br/>
Customer Last Name <input type = "text" name="CUS_LNAME"/><br/>
Customer Gender (M or F) <input type = "text" name="CUS_GENDER"/><br/>
Customer Date of Birth (YYYYMMDD) <input type = "text" name="CUS_DOB"/><br/>
<input type= "submit" name= "add" value= "Add Customer"/>


</form>
</body>
</html>
```

Product Web Page

```
<html>
<title>Northeastern Goods</title>

<!-- header -->
<head>
<div align="center">
<h1>Northeastern Goods                  <img src="NEU-logo.png" alt="NEU-logo" height="90" width="90"></h1><font color="red">For Northeastern Students, By Northeastern Students</font>
</div>
</head>

<!-- tabs -->
<body style = "background-color:#ADD8E6">
</br>
<table border="1" width="100%" ID="Tabs">
<tr>
<th><a href="Central Web Page.html"><h3>Home</h3></a></th>
<th><a href="Employees.html"><h3>Employees</h3></a></th>
<th><a href="Customers.html"><h3>Customers</h3></a></th>
<th><a href="Products.html"><h3>Products</h3></a></th>
<th><a href="Managers.html"><h3>Managers</h3></a></th>
</tr>
</table>

<!-- Welcome Message -->
<h2>You are at the Product Page</h2>
<p style="margin-left: 0px">You can do a few things. </p>
<ul>
<li> Add products to the database </li>
<li> Gather a report of the products </li>
<li> Find the most expensive product </li>
<li> Find the cheapest product </li>
</ul>
<!-- Forms -->

<!-- Add Product -->
<h3>Add Product</h3>
<form method = "post" action = "Add Products.psp">
Product Name <input type= "text" name="PROD_NAME"/><br/>
Product Price <input type = "text" name="PROD_PRICE"/><br/>
<input type= "submit" name= "add" value= "Add Product"/>
</form>

<!-- Product Report + Cheapest/Most expensive product -->
<h3>Get Product Report</h3>
<form method="post" action="Product Report.psp">
<input type='submit' name='report' value='Get Product Report'/>
</form>

<form method = "post" action = "Product Prices.psp">
<h3>When you click this button you will find the cheapest and the most expensive product</h3>
<input type = "submit" name="add" value="Cheapest and Most Expensive Product">
</form>

</body>
</html>
```

Northeastern Goods

For Northeastern Students, By Northeastern Students

Home	Employees	Customers	Products	Managers

You are at the Product Page

You can do a few things.

- Add products to the database
- Gather a report of the products
- Find the most expensive product
- Find the cheapest product

Add Product

Product Name
Product Price
Add Product

Get Product Report

Get Product Report

When you click this button you will find the cheapest and the most expensive product

Cheapest and Most Expensive Product

Manager Web Page

```
<html>
<title>Northeastern Goods</title>

<!-- header -->
<head>
<div align="center">
<h1>Northeastern Goods                <img src="NEU-logo.png" alt="NEU-logo" height="90" width="90"></h1><font color="red">For Northeastern Students, By Northeastern Students</font>
</div>
</head>

<!-- tabs -->
<body style = "background-color:#ADD8E6">
</br>
<table border="1" width="100%" ID="Tabs">
<tr>
<th><a href="Central Web Page.html"><h3>Home</h3></a></th>
<th><a href="Employees.html"><h3>Employees</h3></a></th>
<th><a href="Customers.html"><h3>Customers</h3></a></th>
<th><a href="Products.html"><h3>Products</h3></a></th>
<th><a href="Managers.html"><h3>Managers</h3></a></th>
</tr>
</table>

<!-- Welcome Message -->
<h2>You are at the Manager Page</h2>
<p style="margin-left: 0px">You can get a report for the managers of Northeastern Goods if necessary. </p>

<!-- Manager report -->
<h3>Get Manager Report</h3>
<form method="post" action="Manager Report.psp">
<input type='submit' name='report' value='Get Manager Report'/>
</form>
</body>
</html>
```

Northeastern Goods

For Northeastern Students, By Northeastern Students

Home	Employees	Customers	Products	Managers

You are at the Manager Page

You can get a report for the managers of Northeastern Goods if necessary.

Get Manager Report

Get Manager Report

3. HTML elements
    a. Text

See any of the pages above and the respective HTML codes for text

    b. List

See the Product and Employee pages and the respective HTML codes

    c. Link

The tabs are all linked with each other with the “a href” syntax. See any of the pages above and the respective HTML codes.

    d. Color

Every page has a blue background color. See any of the pages above and the respective HTML codes

    e. Image

Every page has a Northeastern logo at the heading. See any of the pages above and the respective HTML codes

    f. Table

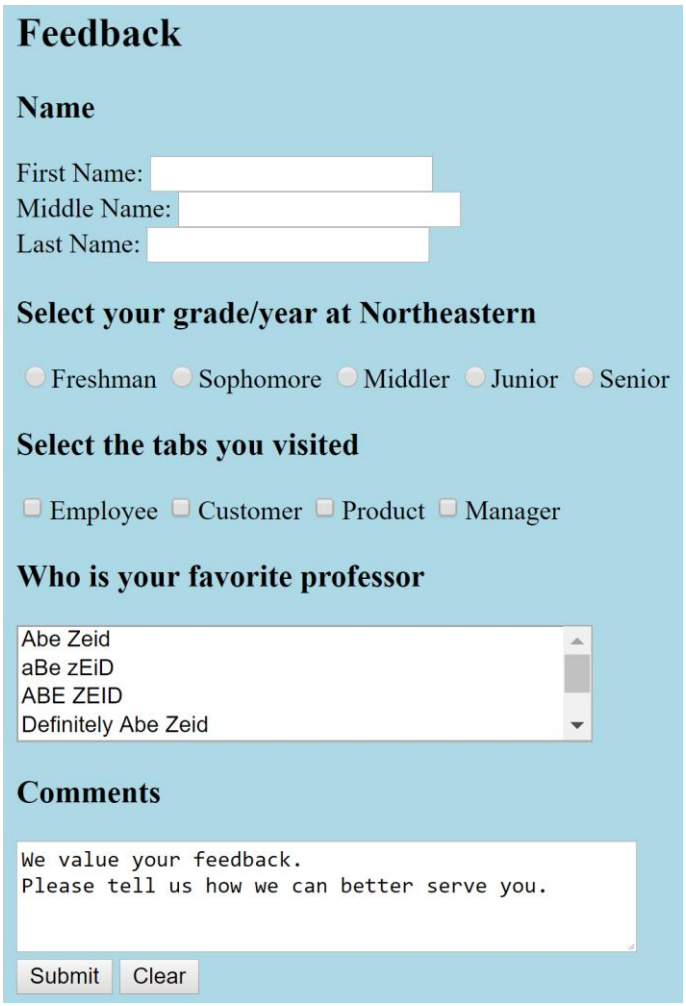
The tabs are aligned with a table (every page). Additionally, the reports (Product and Manager pages) are organized with tables.

    g. Form

There’s a form on every page. See any of the pages above and the respective HTML codes.

4. Form elements – **see Feedback section on the Home Page**
    a. Textbox
    b. Checkbox
    c. Radio
    d. Dropdown list
    e. Text area
    f. Submit
    g. Reset

```
<html>
<head>
 <title>Feedback</title>
 </head>
<body>
<%
 #get all name/value pairs, one by one, from web page
 fName = form.get("fName", "Error in first name")
 mName = form.get("mName", "Error in middle name")
 lName = form.get("lName", "Error in last name")
 year = form.get("year", "Error in year")
 tab = form.get("tab", "Error in tab")
 AbeZeid = form.get("AbeZeid", "Error in Zeid")
 feedback = form.get("feedback", "Error in feedback")

 req.write("<h2>Feedback</h2>")
 req.write("First Name: " + fName + "<br/>")
 req.write("Middle Name: " + mName + "<br/>")
 req.write("Last Name: " + lName + "<br/>")
 req.write("year: " + year + "<br/>")
 req.write("Tab(s): " + tab + "<br/>")
 req.write("Zeid: " + AbeZeid + "<br/>")
 req.write("Feedback: " + feedback + "<br/>")
 %>
 </body>
 </html>
```



(see Home Page HTML code above)

5. Convert Access forms to HTML forms to add data to your MYSQL DB – Lab 22

Add Customer

```
<%
import mysql.connector

#CONNECTING TO THE MySQL SERVER
cnx = mysql.connector.connect(host='instruct.coe.neu.edu', user='3425jlian', password='havepsy', database='3425jlian')
cursor = cnx.cursor()

#get all name/value pairs, one by one, from web page
CUS_FNAME = form.get("CUS_FNAME", "Error in first name")
CUS_LNAME = form.get("CUS_LNAME", "Error in last name")
CUS_GENDER = form.get("CUS_GENDER", "Error in gender")
CUS_DOB = form.get("CUS_DOB", "Error in date of hire")


#convert to strings
CUS_FNAME = str(CUS_FNAME)
CUS_LNAME = str(CUS_LNAME)
CUS_GENDER = str(CUS_GENDER)
CUS_DOB = str(CUS_DOB)

query = "SELECT max(CUS_ID) FROM CUSTOMER;"
cursor.execute(query)

result = cursor.fetchone() #there is fetchall() function also
newPK = "{}".format(result[0] + 1)

insert1="INSERT INTO CUSTOMER(CUS_ID, CUS_FNAME, CUS_LNAME, CUS_GENDER, CUS_DOB) values (%s, %s, %s, %s, %s)"
data = (newPK, CUS_FNAME, CUS_LNAME, CUS_GENDER, CUS_DOB)
cursor.execute(insert1, data)
cnx.commit()
cursor.close()
cnx.close()
%>
```

Add Customer

Customer First Name: Justin
Customer Last Name: Streicher
Customer Gender (M or F): M
Customer Date of Birth (YYYYMMDD): 19981122
Add Customer

```
mysql> SELECT * FROM CUSTOMER;
+--------+-----------+-----------+------------+------------+
| CUS_ID | CUS_FNAME | CUS_LNAME | CUS_GENDER | CUS_DOB    |
+--------+-----------+-----------+------------+------------+
|      1 | Elena     | Shelton   | F          | 1999-02-01 |
|      2 | Reese     | Lindsey   | M          | 1999-06-20 |
|      3 | Heather   | Klein     | F          | 1998-08-07 |
|      4 | Neil      | McCann    | M          | 1999-04-21 |
|      5 | Josh      | Hooper    | M          | 1997-05-02 |
|      6 | Demarcus  | Moody     | M          | 2000-10-31 |
|      7 | Ayaan     | Galvan    | F          | 1996-07-25 |
|      8 | Ishaan    | Oneill    | M          | 1999-11-05 |
|      9 | Araceli   | Floyd     | F          | 1999-01-18 |
|     10 | Carter    | Weeks     | M          | 2000-03-06 |
|     11 | Justin    | Streicher | M          | 1998-11-22 |
+--------+-----------+-----------+------------+------------+
11 rows in set (0.00 sec)
```

Add Product

```
<%
import mysql.connector
#CONNECTING TO THE MySQL SERVER
cnx = mysql.connector.connect(host='instruct.coe.neu.edu', user='3425jlian', password='havepsy', database='3425jlian')
cursor = cnx.cursor()
#get form input
#get all name/value pairs, one by one, from web page
PROD_NAME = form.get("PROD_NAME", "Error in product name")
PROD_PRICE = form.get("PROD_PRICE", "Error in product price")

#convert to strings
PROD_NAME = str(PROD_NAME)
PROD_PRICE = str(PROD_PRICE)

query = "SELECT max(PROD_CODE) FROM PRODUCT;"
cursor.execute(query)

result = cursor.fetchone() #there is fetchall() function also
newPK = "{}".format(result[0] + 1)

insert1="INSERT INTO PRODUCT(PROD_CODE, PROD_NAME, PROD_PRICE) values (%s, %s, %s)"
data = (newPK, PROD_NAME, PROD_PRICE)
cursor.execute(insert1, data)
cnx.commit()
cursor.close()
cnx.close()
%>
```

Add Product

Product Name Kit Kat
Product Price 2.00
Add Product

```
mysql> SELECT * FROM PRODUCT;
+-----------+--------------------+------------+
| PROD_CODE | PROD_NAME          | PROD_PRICE |
+-----------+--------------------+------------+
|         1 | Trail Mix          |       1.75 |
|         2 | Pack of Starburst  |       1.00 |
|         3 | Mechanical Pencils |       3.00 |
|         4 | Ballpoint Pens     |       3.50 |
|         5 | Spiral Notebook    |       2.75 |
|         6 | Looseleaf Paper    |       2.50 |
|         7 | Plastic Folder     |       3.25 |
|         8 | Coffee             |       1.75 |
|         9 | Orange Juice       |       2.00 |
|        10 | Water              |       1.00 |
|        11 | Kit Kat            |       2.00 |
+-----------+--------------------+------------+
11 rows in set (0.00 sec)
```

Add Employee

Add Employee

Employee First Name Anna
Employee Last Name Beck
Employee Date of Birth (YYYYMMDD) 19991210
Employee Date of Hire (YYYYMMDD) 20181210
Employee Gender (M or F) F
Add Employee

```
<%
import mysql.connector

#CONNECTING TO THE MySQL SERVER
cnx = mysql.connector.connect(host='instruct.coe.neu.edu', user='3425jlian', password='havepsy', database='3425jlian')
cursor = cnx.cursor()

#get all name/value pairs, one by one, from web page
EMP_FNAME = form.get("EMP_FNAME", "Error in first name")
EMP_LNAME = form.get("EMP_LNAME", "Error in last name")
EMP_DOB = form.get("EMP_DOB", "Error in date of birth")
EMP_HIRE = form.get("EMP_HIRE", "Error in date of hire")
EMP_GENDER = form.get("EMP_GENDER", "Error in gender")

#convert to strings
EMP_FNAME = str(EMP_FNAME)
EMP_LNAME = str(EMP_LNAME)
EMP_DOB = str(EMP_DOB)
EMP_HIRE = str(EMP_HIRE)
EMP_GENDER = str(EMP_GENDER)

query = "SELECT max(EMP_ID) FROM EMPLOYEE;"
cursor.execute(query)

result = cursor.fetchone() #there is fetchall() function also
newPK = "{}".format(result[0] + 1)

insert1="INSERT INTO EMPLOYEE(EMP_ID, EMP_FNAME, EMP_LNAME, EMP_DOB, EMP_HIRE, EMP_GENDER) values (%s, %s, %s, %s, %s, %s)"
data = (newPK, EMP_FNAME, EMP_LNAME, EMP_DOB, EMP_HIRE, EMP_GENDER)
cursor.execute(insert1, data)
cnx.commit()
cursor.close()
cnx.close()
%>
```

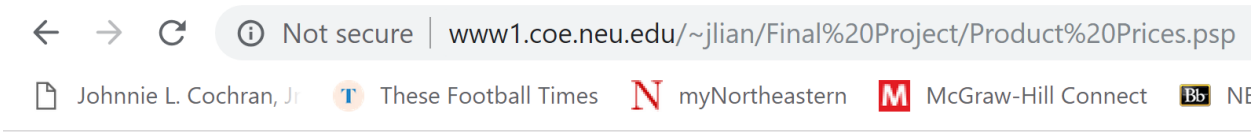
```
mysql> SELECT * FROM EMPLOYEE;
+--------+-------------+-----------+------------+------------+------------+
| EMP_ID | EMP_FNAME   | EMP_LNAME | EMP_DOB    | EMP_HIRE   | EMP_GENDER |
+--------+-------------+-----------+------------+------------+------------+
|      1 | Thomas      | Minutella | 1998-06-22 | 2018-01-02 | M          |
|      2 | Mason       | Kufs      | 1999-06-17 | 2018-01-02 | M          |
|      3 | Jason       | Harris    | 1997-12-11 | 2018-01-10 | M          |
|      4 | Charlie     | Peters    | 1996-09-27 | 2018-02-16 | M          |
|      5 | Max-Antoine | Bonnet    | 1999-01-27 | 2018-02-27 | M          |
|      6 | Matthew     | Thalmann  | 1998-06-28 | 2018-03-01 | M          |
|      7 | Max         | Corkran   | 1999-01-11 | 2018-03-15 | M          |
|      8 | Kelly       | Marchese  | 1999-07-28 | 2018-03-23 | F          |
|      9 | Caleb       | Driesman  | 1999-03-14 | 2018-04-16 | M          |
|     10 | Justin      | Adams     | 1997-11-19 | 2018-04-20 | M          |
|     11 | Jordan      | Lian      | 1999-06-19 | 2018-01-01 | M          |
|     12 | Abe         | Zeid      | 1990-12-25 | 2018-01-02 | M          |
|     13 | Bob         | Ross      | 1942-10-19 | 2018-01-03 | M          |
|     14 | Anna        | Beck      | 1999-12-10 | 2018-12-10 | F          |
+--------+-------------+-----------+------------+------------+------------+
14 rows in set (0.00 sec)
```

6. Convert Access forms to HTML forms to extract info from your MYSQL DB – Lab 22

Cheapest and Most Expensive Product

When you click this button you will find the cheapest and the most expensive product

Cheapest and Most Expensive Product



The most cheapest product offered is $1.00
The most expensive product offered is $3.50

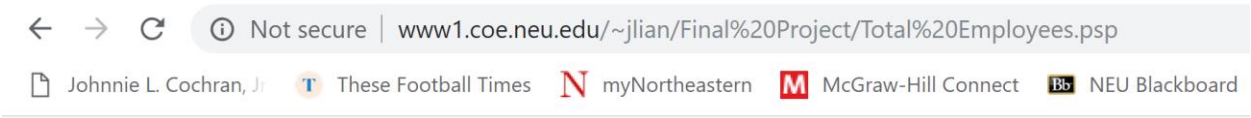
```
<%
import mysql.connector

#CONNECTING TO THE MySQL SERVER
cnx = mysql.connector.connect(host='instruct.coe.neu.edu', user='3425jlian', password='havepsy', database='3425jlian')
cursor = cnx.cursor()

query1 = "SELECT min(PROD_PRICE) FROM PRODUCT;"
query2 = "SELECT max(PROD_PRICE) FROM PRODUCT;"

#query1
cursor.execute(query1)
result1 = cursor.fetchone()
result1 = "{}".format(result1[0])

#query2
cursor.execute(query2)
result2 = cursor.fetchone()
result2 = "{}".format(result2[0])

#print
req.write("<b> The most cheapest product offered is $" + str(result1) + "</b></br>")
req.write("<b> The most expensive product offered is $" + str(result2) + "</b></br>")

%>
```

Total Number of Employees

```
<%
import mysql.connector

#CONNECTING TO THE MySQL SERVER
cnx = mysql.connector.connect(host='instruct.coe.neu.edu', user='3425jlian', password='havepsy', database='3425jlian')
cursor = cnx.cursor()

query1 = "SELECT count(EMP_ID) FROM EMPLOYEE;"

#query1
cursor.execute(query1)
result1 = cursor.fetchone()
result1 = "{}".format(result1[0])

#print
req.write("<b> The number of employees working for the business is " + str(result1) + "</b>")

%>
```

When you click this button you will find how many total employees there are

Total Employees



The number of employees working for the business is 14

7. Convert Access forms to HTML reports; use Web page to access any report – Lab 23

Product Report

Product Report

Product Code	Product Name	Product Price
1	Trail Mix	1.75
2	Pack of Starburst	1.00
3	Mechanical Pencils	3.00
4	Ballpoint Pens	3.50
5	Spiral Notebook	2.75
6	Looseleaf Paper	2.50
7	Plastic Folder	3.25
8	Coffee	1.75
9	Orange Juice	2.00
10	Water	1.00
11	Kit Kat	2.00

Get Product Report

Get Product Report

```
<%
import mysql.connector
#CONNECTING TO THE MySQL SERVER
cnx = mysql.connector.connect(user='3425jlian',password='havepsy', host='instruct.coe.neu.edu', database='3425jlian')
cursor = cnx.cursor()

#Get product records
query="SELECT PROD_CODE, PROD_NAME, PROD_PRICE FROM PRODUCT;"
cursor.execute(query)

#Output results to Web page
req.write("<h3>Product Report</h3>")
req.write("<table border='2'>")
req.write("<tr>")
req.write("<th>Product Code</th>")
req.write("<th>Product Name</th>")
req.write("<th>Product Price</th>")
req.write("</tr>")

#output results to the Web page
for (PROD_CODE, PROD_NAME, PROD_PRICE) in cursor:
    req.write("<tr>")
    req.write ("<td> {} </td>".format(PROD_CODE))
    req.write ("<td> {} </td>".format(PROD_NAME))
    req.write ("<td> {} </td>".format(PROD_PRICE))
req.write("</tr>")
req.write("</table>")
cursor.close()
cnx.close()
%>
```

Manager Report

Manager Report

Manager ID	Mananger First Name	Manager Last Name	Manager Date of Birth	Manager Date of Hire	Manager Gender	Employee ID
1	Jordan	Lian	1999-06-19	2018-01-01	M	11
2	Abe	Zeid	1990-12-25	2018-01-02	M	12
3	Bob	Ross	1942-10-19	2018-01-03	M	13

Get Manager Report

Get Manager Report

```
<%
import mysql.connector
#CONNECTING TO THE MySQL SERVER
cnx = mysql.connector.connect(user='3425jlian',password='havepsy', host='instruct.coe.neu.edu', database='3425jlian')
cursor = cnx.cursor()

#Get product records
query="SELECT MAN_ID, MAN_FNAME, MAN_LNAME, MAN_DOB, MAN_HIRE, MAN_GENDER, EMP_ID FROM MANAGER;"
cursor.execute(query)

#Output results to Web page
req.write("<h3>Manager Report</h3>")
req.write("<table border='2'>")
req.write("<tr>")
req.write("<th>Manager ID</th>")
req.write("<th>Mananger First Name</th>")
req.write("<th>Manager Last Name</th>")
req.write("<th>Manager Date of Birth</th>")
req.write("<th>Manager Date of Hire</th>")
req.write("<th>Manager Gender</th>")
req.write("<th>Employee ID</th>")
req.write("</tr>")

#output results to the Web page
for (MAN_ID, MAN_FNAME, MAN_LNAME, MAN_DOB, MAN_HIRE, MAN_GENDER, EMP_ID) in cursor:
    req.write("<tr>")
    req.write ("<td> {} </td>".format(MAN_ID))
    req.write ("<td> {} </td>".format(MAN_FNAME))
    req.write ("<td> {} </td>".format(MAN_LNAME))
    req.write ("<td> {} </td>".format(MAN_DOB))
    req.write ("<td> {} </td>".format(MAN_HIRE))
    req.write ("<td> {} </td>".format(MAN_GENDER))
    req.write ("<td> {} </td>".format(EMP_ID))
req.write("</tr>")
req.write("</table>")
cursor.close()
cnx.close()
%>
```

8. Create all Python code and SQL queries that support the GUI: screenshots of all Python showing SQL queries. State if screenshots in items 1-7 above; else provide

```
<%
import mysql.connector
#CONNECTING TO THE MySQL SERVER
cnx = mysql.connector.connect(user='3425jlian',password='havepsy', host='instruct.coe.neu.edu', database='3425jlian')
cursor = cnx.cursor()

#Get product records
query="SELECT PROD_CODE, PROD_NAME, PROD_PRICE FROM PRODUCT;"
cursor.execute(query)

#Output results to Web page
req.write("<h3>Product Report</h3>")
req.write("<table border='2'>")
req.write("<tr>")
req.write("<th>Product Code</th>")
req.write("<th>Product Name</th>")
req.write("<th>Product Price</th>")
req.write("</tr>")

#output results to the Web page
for (PROD_CODE, PROD_NAME, PROD_PRICE) in cursor:
    req.write("<tr>")
    req.write ("<td> {} </td>".format(PROD_CODE))
    req.write ("<td> {} </td>".format(PROD_NAME))
    req.write ("<td> {} </td>".format(PROD_PRICE))
req.write("</tr>")
req.write("</table>")
cursor.close()
cnx.close()
%>
```

Product Report .psp file

```
<%
import mysql.connector

#CONNECTING TO THE MySQL SERVER
cnx = mysql.connector.connect(host='instruct.coe.neu.edu', user='3425jlian', password='havepsy', database='3425jlian')
cursor = cnx.cursor()

query1 = "SELECT min(PROD_PRICE) FROM PRODUCT;"
query2 = "SELECT max(PROD_PRICE) FROM PRODUCT;"

#query1
cursor.execute(query1)
result1 = cursor.fetchone()
result1 = "{}".format(result1[0])

#query2
cursor.execute(query2)
result2 = cursor.fetchone()
result2 = "{}".format(result2[0])

#print
req.write("<b> The most cheapest product offered is $" + str(result1) + "</b></br>")
req.write("<b> The most expensive product offered is $" + str(result2) + "</b></br>")

%>
```

Product Price .psp file

```
<%
import mysql.connector
#CONNECTING TO THE MySQL SERVER
cnx = mysql.connector.connect(user='3425jlian',password='havepsy', host='instruct.coe.neu.edu', database='3425jlian')
cursor = cnx.cursor()

#Get product records
query="SELECT MAN_ID, MAN_FNAME, MAN_LNAME, MAN_DOB, MAN_HIRE, MAN_GENDER, EMP_ID FROM MANAGER;"
cursor.execute(query)

#Output results to Web page
req.write("<h3>Manager Report</h3>")
req.write("<table border='2'>")
req.write("<tr>")
req.write("<th>Manager ID</th>")
req.write("<th>Mananger First Name</th>")
req.write("<th>Manager Last Name</th>")
req.write("<th>Manager Date of Birth</th>")
req.write("<th>Manager Date of Hire</th>")
req.write("<th>Manager Gender</th>")
req.write("<th>Employee ID</th>")
req.write("</tr>")

#output results to the Web page
for (MAN_ID, MAN_FNAME, MAN_LNAME, MAN_DOB, MAN_HIRE, MAN_GENDER, EMP_ID) in cursor:
    req.write("<tr>")
    req.write ("<td> {} </td>".format(MAN_ID))
    req.write ("<td> {} </td>".format(MAN_FNAME))
    req.write ("<td> {} </td>".format(MAN_LNAME))
    req.write ("<td> {} </td>".format(MAN_DOB))
    req.write ("<td> {} </td>".format(MAN_HIRE))
    req.write ("<td> {} </td>".format(MAN_GENDER))
    req.write ("<td> {} </td>".format(EMP_ID))
req.write("</tr>")
req.write("</table>")
cursor.close()
cnx.close()
%>
```

Manager Report .psp file

```
<html>
<head>
 <title>Feedback</title>
</head>
<body>
<%
 #get all name/value pairs, one by one, from web page
 fName = form.get("fName", "Error in first name")
 mName = form.get("mName", "Error in middle name")
 lName = form.get("lName", "Error in last name")
 year = form.get("year", "Error in year")
 tab = form.get("tab", "Error in tab")
 AbeZeid = form.get("AbeZeid", "Error in Zeid")
 feedback = form.get("feedback", "Error in feedback")

 req.write("<h2>Feedback</h2>")
 req.write("First Name: " + fName + "<br/>")
 req.write("Middle Name: " + mName + "<br/>")
 req.write("Last Name: " + lName + "<br/>")
 req.write("year: " + year + "<br/>")
 req.write("Tab(s): " + tab + "<br/>")
 req.write("Zeid: " + AbeZeid + "<br/>")
 req.write("Feedback: " + feedback + "<br/>")
%>
</body>
</html>
```

Feedback .psp file

```
<%
import mysql.connector
#CONNECTING TO THE MySQL SERVER
cnx = mysql.connector.connect(host='instruct.coe.neu.edu', user='3425jlian', password='havepsy', database='3425jlian')
cursor = cnx.cursor()
#get form input
#get all name/value pairs, one by one, from web page
PROD_NAME = form.get("PROD_NAME", "Error in product name")
PROD_PRICE = form.get("PROD_PRICE", "Error in product price")

#convert to strings
PROD_NAME = str(PROD_NAME)
PROD_PRICE = str(PROD_PRICE)

query = "SELECT max(PROD_CODE) FROM PRODUCT;"
cursor.execute(query)

result = cursor.fetchone() #there is fetchall() function also
newPK = "{}".format(result[0] + 1)

insert1="INSERT INTO PRODUCT(PROD_CODE, PROD_NAME, PROD_PRICE) values (%s, %s, %s)"
data = (newPK, PROD_NAME, PROD_PRICE)
cursor.execute(insert1, data)
cnx.commit()
cursor.close()
cnx.close()
%>
```

Add Product .psp file

```
<%
import mysql.connector

#CONNECTING TO THE MySQL SERVER
cnx = mysql.connector.connect(host='instruct.coe.neu.edu', user='3425jlian', password='havepsy', database='3425jlian')
cursor = cnx.cursor()

#get all name/value pairs, one by one, from web page
EMP_FNAME = form.get("EMP_FNAME", "Error in first name")
EMP_LNAME = form.get("EMP_LNAME", "Error in last name")
EMP_DOB = form.get("EMP_DOB", "Error in date of birth")
EMP_HIRE = form.get("EMP_HIRE", "Error in date of hire")
EMP_GENDER = form.get("EMP_GENDER", "Error in gender")

#convert to strings
EMP_FNAME = str(EMP_FNAME)
EMP_LNAME = str(EMP_LNAME)
EMP_DOB = str(EMP_DOB)
EMP_HIRE = str(EMP_HIRE)
EMP_GENDER = str(EMP_GENDER)

query = "SELECT max(EMP_ID) FROM EMPLOYEE;"
cursor.execute(query)

result = cursor.fetchone() #there is fetchall() function also
newPK = "{}".format(result[0] + 1)

insert1="INSERT INTO EMPLOYEE(EMP_ID, EMP_FNAME, EMP_LNAME, EMP_DOB, EMP_HIRE, EMP_GENDER) values (%s, %s, %s, %s, %s, %s)"
data = (newPK, EMP_FNAME, EMP_LNAME, EMP_DOB, EMP_HIRE, EMP_GENDER)
cursor.execute(insert1, data)
cnx.commit()
cursor.close()
cnx.close()
%>
```

Add Employee .psp file

```
<%
import mysql.connector

#CONNECTING TO THE MySQL SERVER
cnx = mysql.connector.connect(host='instruct.coe.neu.edu', user='3425jlian', password='havepsy', database='3425jlian')
cursor = cnx.cursor()

#get all name/value pairs, one by one, from web page
CUS_FNAME = form.get("CUS_FNAME", "Error in first name")
CUS_LNAME = form.get("CUS_LNAME", "Error in last name")
CUS_GENDER = form.get("CUS_GENDER", "Error in gender")
CUS_DOB = form.get("CUS_DOB", "Error in date of hire")


#convert to strings
CUS_FNAME = str(CUS_FNAME)
CUS_LNAME = str(CUS_LNAME)
CUS_GENDER = str(CUS_GENDER)
CUS_DOB = str(CUS_DOB)

query = "SELECT max(CUS_ID) FROM CUSTOMER;"
cursor.execute(query)

result = cursor.fetchone() #there is fetchall() function also
newPK = "{}".format(result[0] + 1)

insert1="INSERT INTO CUSTOMER(CUS_ID, CUS_FNAME, CUS_LNAME, CUS_GENDER, CUS_DOB) values (%s, %s, %s, %s, %s)"
data = (newPK, CUS_FNAME, CUS_LNAME, CUS_GENDER, CUS_DOB)
cursor.execute(insert1, data)
cnx.commit()
cursor.close()
cnx.close()
%>
```

Add Customer .psp file

```
<%
import mysql.connector

#CONNECTING TO THE MySQL SERVER
cnx = mysql.connector.connect(host='instruct.coe.neu.edu', user='3425jlian', password='havepsy', database='3425jlian')
cursor = cnx.cursor()

query1 = "SELECT count(EMP_ID) FROM EMPLOYEE;"

#query1
cursor.execute(query1)
result1 = cursor.fetchone()
result1 = "{}".format(result1[0])

#print
req.write("<b> The number of employees working for the business is " + str(result1) + "</b>")

%>
```

Total Employees .psp file

9. Convert your Access use cases to HTML use cases, i.e. use HTML Web pages and supporting Python code: description and necessary screenshots
    - Case 1 – New Employee is Hired

If a new employee is hired, the company needs to add the employee into the database. Here's how we do it.

```
<%
import mysql.connector

#CONNECTING TO THE MySQL SERVER
cnx = mysql.connector.connect(host='instruct.coe.neu.edu', user='3425jlian', password='havepsy', database='3425jlian')
cursor = cnx.cursor()

#get all name/value pairs, one by one, from web page
EMP_FNAME = form.get("EMP_FNAME", "Error in first name")
EMP_LNAME = form.get("EMP_LNAME", "Error in last name")
EMP_DOB = form.get("EMP_DOB", "Error in date of birth")
EMP_HIRE = form.get("EMP_HIRE", "Error in date of hire")
EMP_GENDER = form.get("EMP_GENDER", "Error in gender")

#convert to strings
EMP_FNAME = str(EMP_FNAME)
EMP_LNAME = str(EMP_LNAME)
EMP_DOB = str(EMP_DOB)
EMP_HIRE = str(EMP_HIRE)
EMP_GENDER = str(EMP_GENDER)

query = "SELECT max(EMP_ID) FROM EMPLOYEE;"
cursor.execute(query)

result = cursor.fetchone() #there is fetchall() function also
newPK = "{}".format(result[0] + 1)

insert1="INSERT INTO EMPLOYEE(EMP_ID, EMP_FNAME, EMP_LNAME, EMP_DOB, EMP_HIRE, EMP_GENDER) values (%s, %s, %s, %s, %s, %s)"
data = (newPK, EMP_FNAME, EMP_LNAME, EMP_DOB, EMP_HIRE, EMP_GENDER)
cursor.execute(insert1, data)
cnx.commit()
cursor.close()
cnx.close()
%>
```

Create Python code to output the results

Add Employee

Employee First Name: Anna

Employee Last Name: Beck

Employee Date of Birth (YYYYMMDD): 19991210

Employee Date of Hire (YYYYMMDD): 20181210

Employee Gender (M or F): F

Add Employee

Go to the Employees page and put the data in. Click the Add Employee button to add!

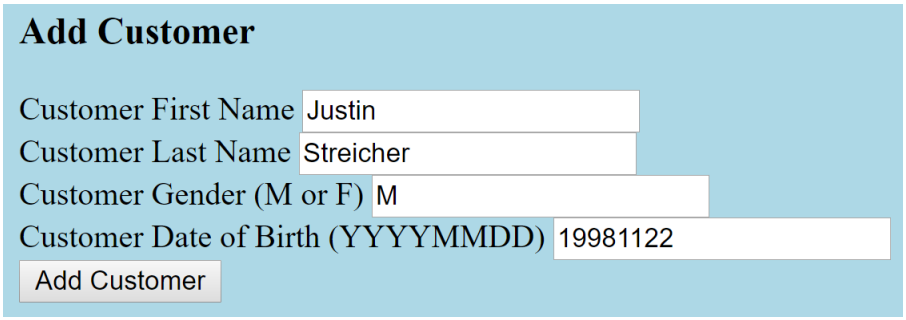
```
mysql> SELECT * FROM EMPLOYEE;
+--------+-------------+-----------+------------+------------+------------+
| EMP_ID | EMP_FNAME   | EMP_LNAME | EMP_DOB    | EMP_HIRE   | EMP_GENDER |
+--------+-------------+-----------+------------+------------+------------+
|      1 | Thomas      | Minutella | 1998-06-22 | 2018-01-02 | M          |
|      2 | Mason       | Kufs      | 1999-06-17 | 2018-01-02 | M          |
|      3 | Jason       | Harris    | 1997-12-11 | 2018-01-10 | M          |
|      4 | Charlie     | Peters    | 1996-09-27 | 2018-02-16 | M          |
|      5 | Max-Antoine | Bonnet    | 1999-01-27 | 2018-02-27 | M          |
|      6 | Matthew     | Thalmann  | 1998-06-28 | 2018-03-01 | M          |
|      7 | Max         | Corkran   | 1999-01-11 | 2018-03-15 | M          |
|      8 | Kelly       | Marchese  | 1999-07-28 | 2018-03-23 | F          |
|      9 | Caleb       | Driesman  | 1999-03-14 | 2018-04-16 | M          |
|     10 | Justin      | Adams     | 1997-11-19 | 2018-04-20 | M          |
|     11 | Jordan      | Lian      | 1999-06-19 | 2018-01-01 | M          |
|     12 | Abe         | Zeid      | 1990-12-25 | 2018-01-02 | M          |
|     13 | Bob         | Ross      | 1942-10-19 | 2018-01-03 | M          |
|     14 | Anna        | Beck      | 1999-12-10 | 2018-12-10 | F          |
+--------+-------------+-----------+------------+------------+------------+
14 rows in set (0.00 sec)
```

Results!

- Case 2 – Customer purchases something from the store, and their name is not in our database

```
<%
import mysql.connector

#CONNECTING TO THE MySQL SERVER
cnx = mysql.connector.connect(host='instruct.coe.neu.edu', user='3425jlian', password='havepsy', database='3425jlian')
cursor = cnx.cursor()

#get all name/value pairs, one by one, from web page
CUS_FNAME = form.get("CUS_FNAME", "Error in first name")
CUS_LNAME = form.get("CUS_LNAME", "Error in last name")
CUS_GENDER = form.get("CUS_GENDER", "Error in gender")
CUS_DOB = form.get("CUS_DOB", "Error in date of hire")


#convert to strings
CUS_FNAME = str(CUS_FNAME)
CUS_LNAME = str(CUS_LNAME)
CUS_GENDER = str(CUS_GENDER)
CUS_DOB = str(CUS_DOB)

query = "SELECT max(CUS_ID) FROM CUSTOMER;"
cursor.execute(query)

result = cursor.fetchone() #there is fetchall() function also
newPK = "{}".format(result[0] + 1)

insert1="INSERT INTO CUSTOMER(CUS_ID, CUS_FNAME, CUS_LNAME, CUS_GENDER, CUS_DOB) values (%s, %s, %s, %s, %s)"
data = (newPK, CUS_FNAME, CUS_LNAME, CUS_GENDER, CUS_DOB)
cursor.execute(insert1, data)
cnx.commit()
cursor.close()
cnx.close()
%>
```

Create the Python code to output the results



Go to the Customer pages and put the data in. Click the Add Customer button to add!

```
mysql> SELECT * FROM CUSTOMER;
+--------+-----------+-----------+------------+------------+
| CUS_ID | CUS_FNAME | CUS_LNAME | CUS_GENDER | CUS_DOB    |
+--------+-----------+-----------+------------+------------+
|      1 | Elena     | Shelton   | F          | 1999-02-01 |
|      2 | Reese     | Lindsey   | M          | 1999-06-20 |
|      3 | Heather   | Klein     | F          | 1998-08-07 |
|      4 | Neil      | McCann    | M          | 1999-04-21 |
|      5 | Josh      | Hooper    | M          | 1997-05-02 |
|      6 | Demarcus  | Moody     | M          | 2000-10-31 |
|      7 | Ayaan     | Galvan    | F          | 1996-07-25 |
|      8 | Ishaan    | Oneill    | M          | 1999-11-05 |
|      9 | Araceli   | Floyd     | F          | 1999-01-18 |
|     10 | Carter    | Weeks     | M          | 2000-03-06 |
|     11 | Justin    | Streicher | M          | 1998-11-22 |
+--------+-----------+-----------+------------+------------+
11 rows in set (0.00 sec)
```

Results!

- Case 3 – Manager needs to submit a report of all the company's employees for the fiscal year 2018.

```
<%
import mysql.connector
#CONNECTING TO THE MySQL SERVER
cnx = mysql.connector.connect(user='3425jlian',password='havepsy', host='instruct.coe.neu.edu', database='3425jlian')
cursor = cnx.cursor()

#Get product records
query="SELECT MAN_ID, MAN_FNAME, MAN_LNAME, MAN_DOB, MAN_HIRE, MAN_GENDER, EMP_ID FROM MANAGER;"
cursor.execute(query)

#Output results to Web page
req.write("<h3>Manager Report</h3>")
req.write("<table border='2'>")
req.write("<tr>")
req.write("<th>Manager ID</th>")
req.write("<th>Mananger First Name</th>")
req.write("<th>Manager Last Name</th>")
req.write("<th>Manager Date of Birth</th>")
req.write("<th>Manager Date of Hire</th>")
req.write("<th>Manager Gender</th>")
req.write("<th>Employee ID</th>")
req.write("</tr>")

#output results to the Web page
for (MAN_ID, MAN_FNAME, MAN_LNAME, MAN_DOB, MAN_HIRE, MAN_GENDER, EMP_ID) in cursor:
    req.write("<tr>")
    req.write ("<td> {} </td>".format(MAN_ID))
    req.write ("<td> {} </td>".format(MAN_FNAME))
    req.write ("<td> {} </td>".format(MAN_LNAME))
    req.write ("<td> {} </td>".format(MAN_DOB))
    req.write ("<td> {} </td>".format(MAN_HIRE))
    req.write ("<td> {} </td>".format(MAN_GENDER))
    req.write ("<td> {} </td>".format(EMP_ID))
req.write("</tr>")
req.write("</table>")
cursor.close()
cnx.close()
%>
```

Create a Python code to output the Manager table from the database

Go to the Managers page and click the Get Manager Report button to get the report!

Manager Report

Manager ID	Mananger First Name	Manager Last Name	Manager Date of Birth	Manager Date of Hire	Manager Gender	Employee ID
1	Jordan	Lian	1999-06-19	2018-01-01	M	11
2	Abe	Zeid	1990-12-25	2018-01-02	M	12
3	Bob	Ross	1942-10-19	2018-01-03	M	13

Results!

10. All Web pages of the project website are accessible through a Web browser

http://www1.coe.neu.edu/~jlian/

click on Final Project and click either one the 5 web pages

- Central Web Page.html
- Managers.html
- Products.html

- Customers.html
- Employees.html

DB Theory Questions)

1. Describe/discuss the expandability/scalability of your DB GUI Design. Is your design robust enough to handle future growth? Offer example of an additional form or report to be added.

My DB GUI Design is expandable and scalable because it is well organized, and the forms are easily put together, so it is easy to add data into the database. The tabs are organized in a table and the HTML code makes it easy to add tabs and forms. Additionally, the web pages are all identically formatted, so it is easy to add pages without difficulty. For instance, I could easily create two forms that would compile reports for the employees and customers and put them into the employee and customer pages respectively.

2. How idiot-proof is your front end? Discuss how your GUI and DB back end work together to prevent users from entering junk into your DB.

My front end is idiot-proof. Every piece of data has a datatype that was done in MYSQL and the .psp output files were written such that error messages would pop up if ill-formatted data was put into the database. Therefore, the user will know if they have entered junk into the database. The Web based GUI will let them know and MYSQL will reject the junk data. If valid data is entered, nothing will show. That means no hitches have occurred and the data is now in the system.

3. How do you compare both GUIs (Access and Web based): which did you enjoy more? Why? Which took more time?

I enjoyed the Web based GUI more because the queries were more applicable, and the GUI loads much faster on the Internet. Additionally, it was much easier to undo and redo mistakes on HTML, .psp, or on MYSQL compared to Access. Access had many restrictions so it was tough to redo mistakes. Often times if you made a mistake on Access you'd either have to work around it or spend a lot of time undoing your mistake which was not productive. As a result, I enjoyed creating the Web based GUI much more than the Access based GUI.

4. Project feedback: write your candid feedback about the project; hours it took, your experience with XHTML and Python, what you learned from the project.

I didn't take that much time to create the Web based database compared to the Access based database. I did initially have trouble uploading the data from Access, but after that, writing the HTML code wasn't too difficult. I linked 4 pages together with identical formats, and I used the Lab assignments to help me create the forms and reports.

Uploading the data was initially a struggle, but I realized my table schema was flawed. I slightly altered the schema, so the data would upload correctly to the MYSQL database, and then I was set. However, the changed table schema did not affect my Web based GUI because I did not want people to be able to alter that set of data on the Internet.

5. Course exit feedback: write your candid feedback about the course. Any suggestions to improve?

I enjoyed the course and I learned a lot of useful things (particularly SQL and MYSQL), but I found that at times Professor Zeid would go astray at times and get mixed up during class. This was not helpful because I was very new to most of this material and some clarity during class would've been more helpful. Regardless, it's been a fun semester, and I've learned a lot. I can't ask for much more, and I hope to work with Professor Zeid in the future.