**Project 1, Item 2 Documentation**

The project has been implemented using PHP programming language and MYSQL database. The front end web interface is a combination of HTML and CSS.

**Tables and Triggers: (Following code has also been attached with the name load\_data.sql)**

create database pro1;

use pro1;

create table login(user varchar(50) not null,password varchar(50) not null,status char(9),md5 varchar(50), primary key(user));

create table employee(user varchar(50) not null,ssn int (10) not null,fname char(10),lname char(10),age int(3),address varchar(50),vnt date,vet date,primary key(ssn));

create table bonus(user varchar(10),vnt datetime,vet datetime,salary int(10),primary key(user,VNT));

create table item(id int(10) not null,name varchar(50),description varchar(50),amount int(10),cost int(10),primary key(id));

create table cost\_log(sno int(10) not null AUTO\_INCREMENT,id int(10),oldcost int(10),newcost int(10),diff int(10),vnt datetime,vet datetime,primary key(sno));

DELIMITER $$

create trigger logus after update on login

for each row

BEGIN

IF new.status = 'active'

THEN

insert into bonus values(new.user,NOW(),'null','null');

ELSEIF new.status = 'inactive'

THEN

update bonus set vet = NOW() where user = new.user;

SET @diff = (SELECT TIMESTAMPDIFF(HOUR,vnt, vet) from bonus where user=new.user and salary is NULL);

SET @whole = @diff \* 6;

update bonus set salary = @whole where user=new.user and salary is NULL;

END IF;

END;

$$

DELIMITER ;

DELIMITER $$

create trigger icos1 after update on item

for each row

BEGIN

IF (NOT EXISTS(select \* from cost\_log where id = new.id))

THEN

insert into cost\_log(id,oldcost,vnt) values(new.id,new.cost,NOW());

ELSEIF (EXISTS(select \* from cost\_log where id = new.id))

THEN

update cost\_log set newcost = new.cost where id = new.id and newcost is NULL;

SET @diff = (select newcost-oldcost from cost\_log where id = new.id and vet is NULL);

update cost\_log set diff = @diff,vet = NOW() where id = new.id and vet is NULL;

insert into cost\_log(id,oldcost,vnt) values(new.id,new.cost,curdate()+1);

END IF;

END;

$$

DELIMITER ;

DELIMITER $$

create trigger icos2 after insert on item

for each row

BEGIN

IF (NOT EXISTS(select \* from cost\_log where id = new.id))

THEN

insert into cost\_log(id,oldcost,vnt) values(new.id,new.cost,NOW());

END IF;

END;

$$

DELIMITER ;

DELIMITER $$

create trigger prerecords after delete on employee

for each row

BEGIN

update employee set vet=CURDATE() where user = old.user;

END;

$$

DELIMITER ;

**Tables Description: (Output file is attached with the name output.log)**

mysql> notee

mysql> source load\_data.sql

Query OK, 1 row affected (0.01 sec)

Database changed

Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.03 sec)

Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.03 sec)

Query OK, 0 rows affected (0.03 sec)

Query OK, 0 rows affected (0.03 sec)

Query OK, 0 rows affected (0.03 sec)

mysql> desc employee;

+---------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+---------+-------------+------+-----+---------+-------+

| user | varchar(50) | NO | | NULL | |

| ssn | int(10) | NO | PRI | NULL | |

| fname | char(10) | YES | | NULL | |

| lname | char(10) | YES | | NULL | |

| age | int(3) | YES | | NULL | |

| address | varchar(50) | YES | | NULL | |

| vnt | datetime | YES | | NULL | |

| vet | datetime | YES | | NULL | |

+---------+-------------+------+-----+---------+-------+

8 rows in set (0.03 sec)

mysql> desc item;

+-------------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+-------------+-------------+------+-----+---------+-------+

| id | int(10) | NO | PRI | NULL | |

| name | varchar(50) | YES | | NULL | |

| description | varchar(50) | YES | | NULL | |

| amount | int(10) | YES | | NULL | |

| cost | int(10) | YES | | NULL | |

+-------------+-------------+------+-----+---------+-------+

5 rows in set (0.03 sec)

mysql> desc login;

+----------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+----------+-------------+------+-----+---------+-------+

| user | varchar(50) | NO | PRI | NULL | |

| password | varchar(50) | NO | | NULL | |

| status | char(9) | YES | | NULL | |

| md5 | varchar(50) | YES | | NULL | |

+----------+-------------+------+-----+---------+-------+

4 rows in set (0.03 sec)

mysql> desc bonus;

+--------+-------------+------+-----+---------------------+-------+

| Field | Type | Null | Key | Default | Extra |

+--------+-------------+------+-----+---------------------+-------+

| user | varchar(10) | NO | PRI | | |

| vnt | datetime | NO | PRI | 0000-00-00 00:00:00 | |

| vet | datetime | YES | | NULL | |

| salary | int(10) | YES | | NULL | |

+--------+-------------+------+-----+---------------------+-------+

4 rows in set (0.03 sec)

mysql> desc cost\_log;

+---------+----------+------+-----+---------+----------------+

| Field | Type | Null | Key | Default | Extra |

+---------+----------+------+-----+---------+----------------+

| sno | int(10) | NO | PRI | NULL | auto\_increment |

| id | int(10) | YES | | NULL | |

| oldcost | int(10) | YES | | NULL | |

| newcost | int(10) | YES | | NULL | |

| diff | int(10) | YES | | NULL | |

| vnt | datetime | YES | | NULL | |

| vet | datetime | YES | | NULL | |

+---------+----------+------+-----+---------+----------------+

7 rows in set (0.03 sec)

mysql> show triggers;

+------------+--------+----------

+-------------------------------------------------------------------

--------------------------------------------------------------------

--------------------------------------------------------------------

--------------------------------------------------------------------

--------------------------------------------------------------------

--------------------------------------------------------------------

--------------------------------------------------------------------

--------------------------------------------------------------------

------------+--------+---------+----------+----------------

+----------------------+----------------------+--------------------+

| Trigger | Event | Table | Statement

| Timing | Created | sql\_mode |

Definer | character\_set\_client | collation\_connection |

Database Collation |

+------------+--------+----------

+-------------------------------------------------------------------

--------------------------------------------------------------------

--------------------------------------------------------------------

--------------------------------------------------------------------

--------------------------------------------------------------------

--------------------------------------------------------------------

--------------------------------------------------------------------

--------------------------------------------------------------------

------------+--------+---------+----------+----------------

+----------------------+----------------------+--------------------+

| prerecords | DELETE | employee | BEGIN

update employee set vet=CURDATE() where user = old.user;

END

| AFTER | NULL | | root@localhost | cp850

| cp850\_general\_ci | latin1\_swedish\_ci |

| icos2 | INSERT | item | BEGIN

IF (NOT EXISTS(select \* from cost\_log where id = new.id))

THEN

insert into cost\_log(id,oldcost,vnt) values(new.id,new.cost,NOW());

END IF;

END

| AFTER | NULL |

| root@localhost | cp850 | cp850\_general\_ci

| latin1\_swedish\_ci |

| icos1 | UPDATE | item | BEGIN

IF (NOT EXISTS(select \* from cost\_log where id = new.id))

THEN

insert into cost\_log(id,oldcost,vnt) values(new.id,new.cost,NOW());

ELSEIF (EXISTS(select \* from cost\_log where id = new.id))

THEN

update cost\_log set newcost = new.cost where id = new.id and newcost

is NULL;

SET @diff = (select newcost-oldcost from cost\_log where id =

new.id and vet is NULL);

update cost\_log set diff = @diff,vet = NOW() where id = new.id and

vet is NULL;

insert into cost\_log(id,oldcost,vnt) values(new.id,new.cost,curdate

()+1);

END IF;

END | AFTER | NULL | | root@localhost | cp850

| cp850\_general\_ci | latin1\_swedish\_ci |

| logus | UPDATE | login | BEGIN

IF new.status = 'active'

THEN

insert into bonus values(new.user,NOW(),'null','null');

ELSEIF new.status = 'inactive'

THEN

update bonus set vet = NOW() where user = new.user;

SET @diff = (SELECT TIMESTAMPDIFF(HOUR,vnt, vet) from bonus

where user=new.user and salary is NULL);

SET @whole = @diff \* 6;

update bonus set salary = @whole where user=new.user and salary is

NULL;

END IF;

END

|

AFTER | NULL | | root@localhost | cp850

| cp850\_general\_ci | latin1\_swedish\_ci |

+------------+--------+----------

+-------------------------------------------------------------------

--------------------------------------------------------------------

--------------------------------------------------------------------

--------------------------------------------------------------------

--------------------------------------------------------------------

--------------------------------------------------------------------

--------------------------------------------------------------------

--------------------------------------------------------------------

------------+--------+---------+----------+----------------

+----------------------+----------------------+--------------------+

4 rows in set (0.02 sec)

mysql> note

**Tables with Data: (The following inputs are available as attachment)**

1. Login.txt
2. Employee.txt
3. Item.txt

**PHP Codes:**

1. Connection.php
2. Addemployee.php
3. Additem.php
4. Analyze.php
5. Costupdate.php
6. Delemployee.php
7. Delitem.php
8. Register.php
9. Searchitem.php
10. Searchemp.php
11. Sigin.php
12. Sigin.html
13. Searchitem.html
14. Searchemp.html
15. Addemployee.html
16. Additem.html
17. Delete.html
18. Index.html
19. Managerview.html
20. Records.html
21. Employee.php
22. Readfile.php
23. Index.css

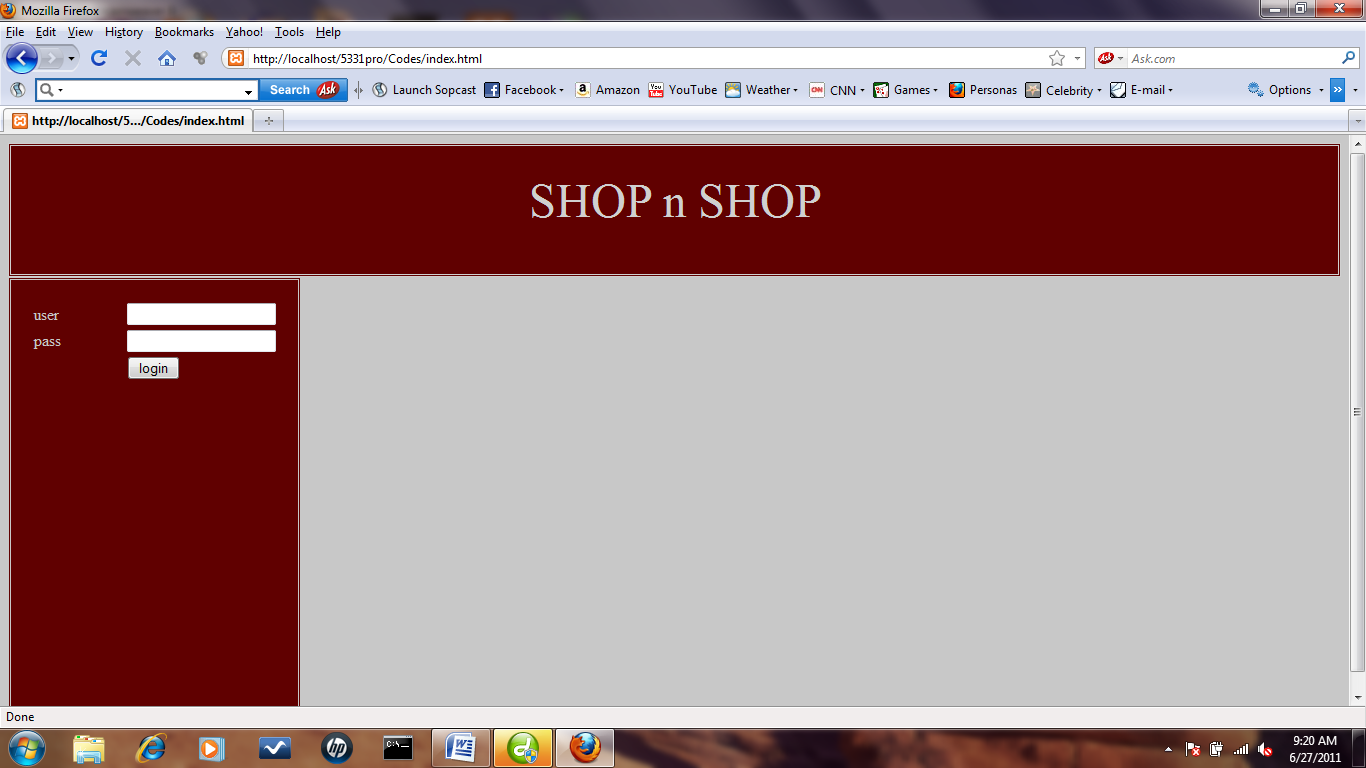
**Some Important Assumptions**

Our database contains the data of a SHOP. At first, to access the database any shop employee has to login through login page. Employees will use already assigned user and password. After login they view the information from the bonus table. When the Manager logs in he would have the options such as add employee, add item, search emplyee, search item and employee prerecords. Manager can add, delete attributes from the employee and item table.

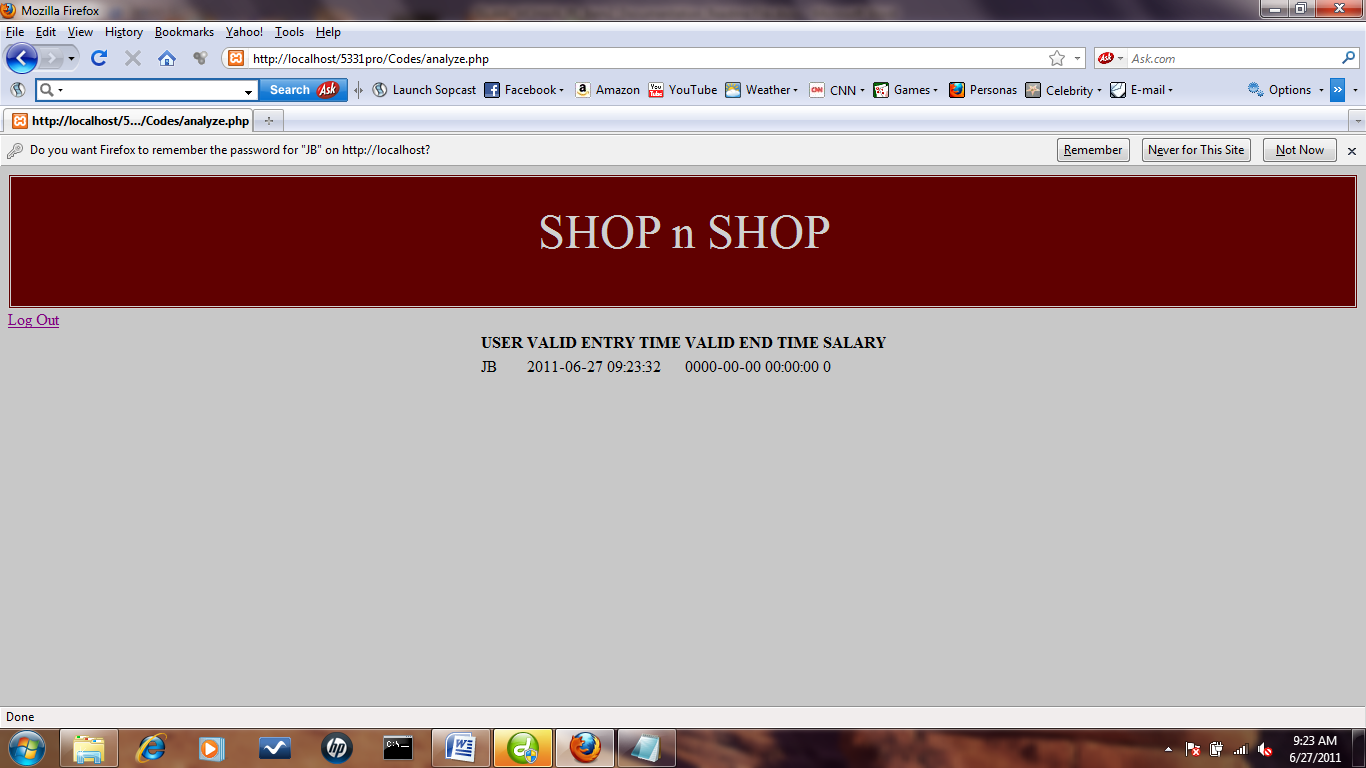
**Description of web application:**

**1.** We have used apache server for our database(WAMP s/w). It must be mentioned that our database name is pro1.

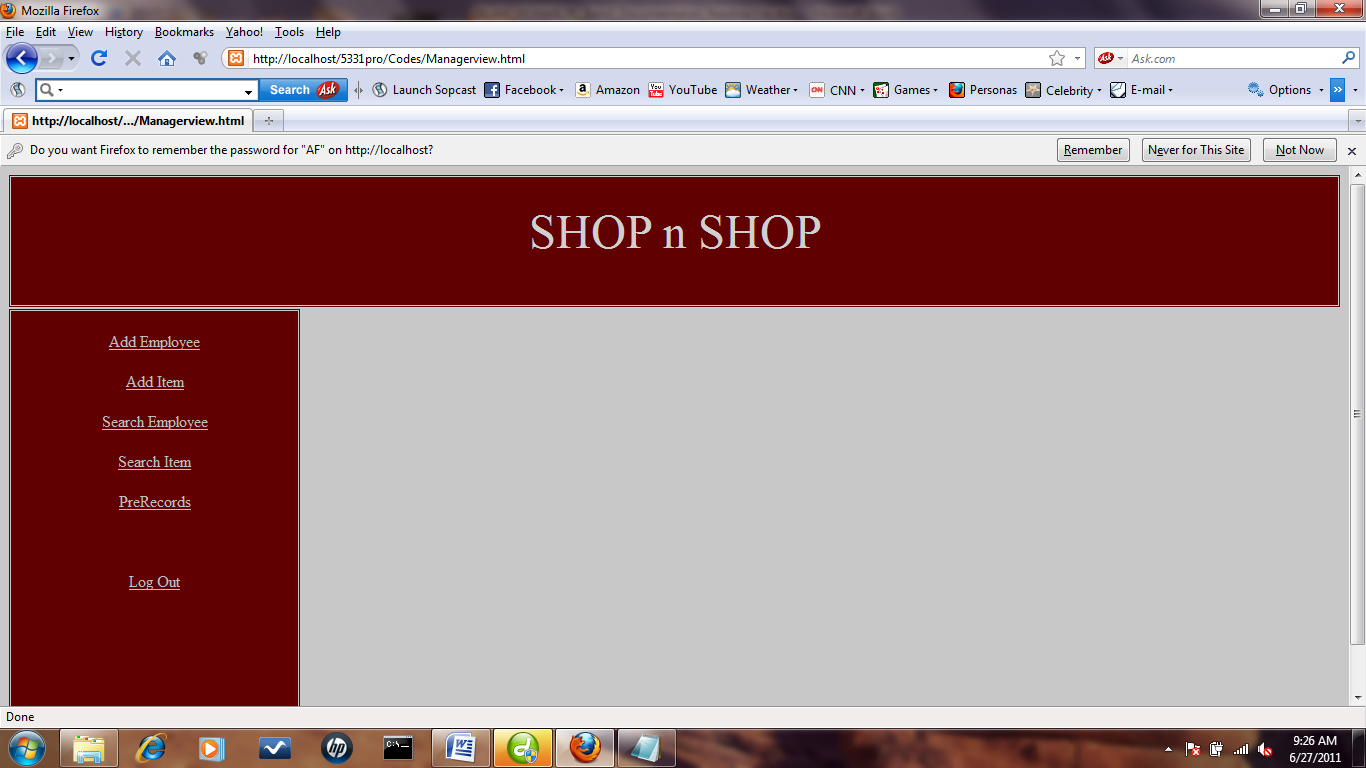
**2.** To get access into the database we need to open index.html which is the login page for anyperon to use the database.



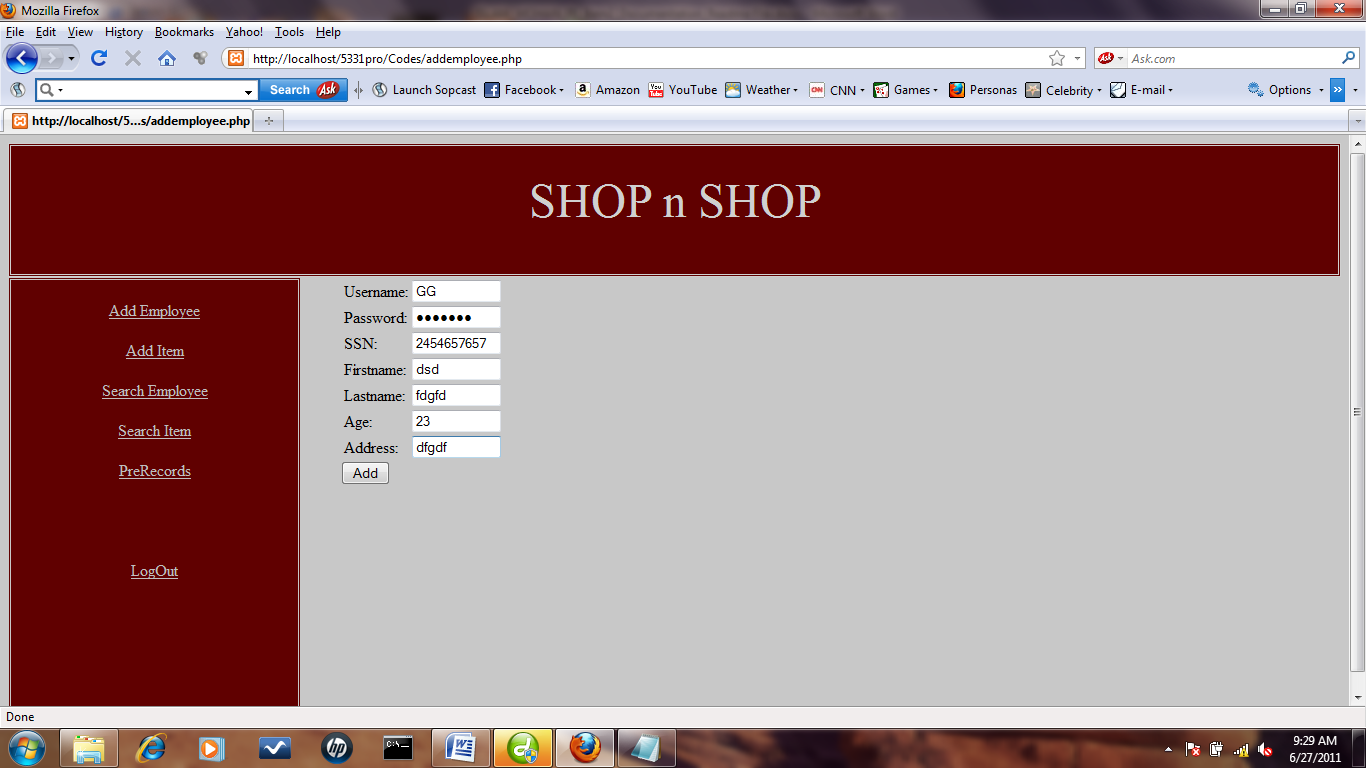
Employees have to login with already given user and password. Now for general employees, they will see their information from the bonus table after they login. Here trigger logus will update information in bonus table when a customer is logged in or status is set to active. The table will also contain the temporal information. This will be displayed in analyze.php page.



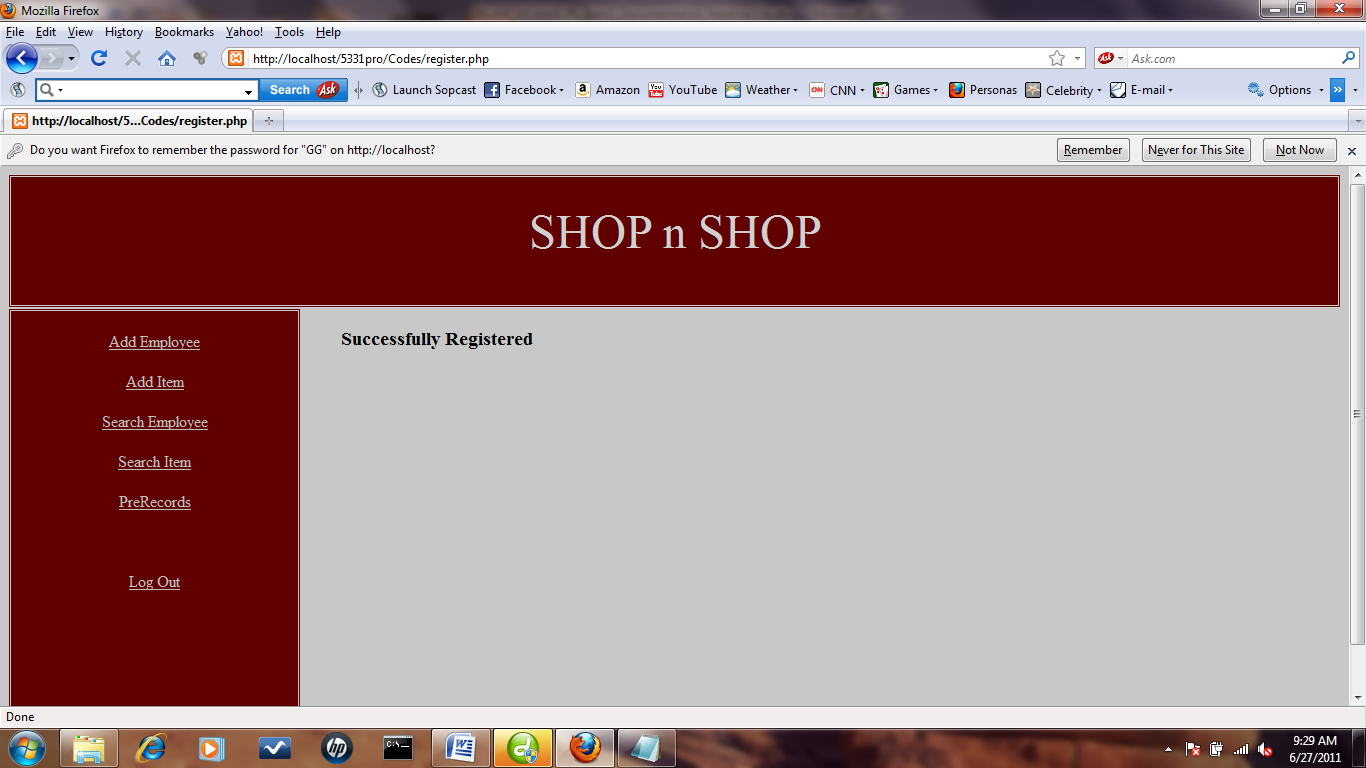
Manager will view a different page after he logs into database. That will be under Managerview.html. Manager will have the option to add and search employee and items.



The next transaction is to add information about a new employee. By clicking the Add Employee link, the following appears.

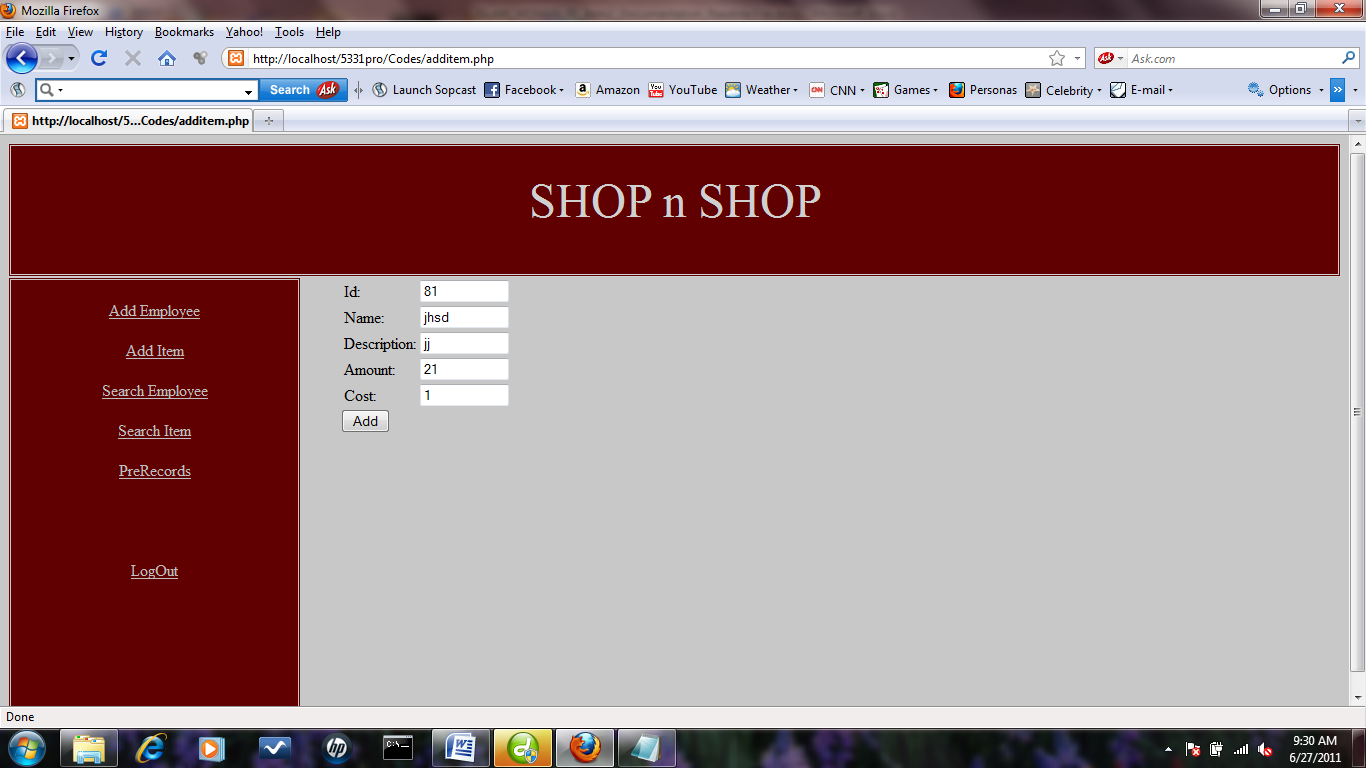


New Employee information is inserted.

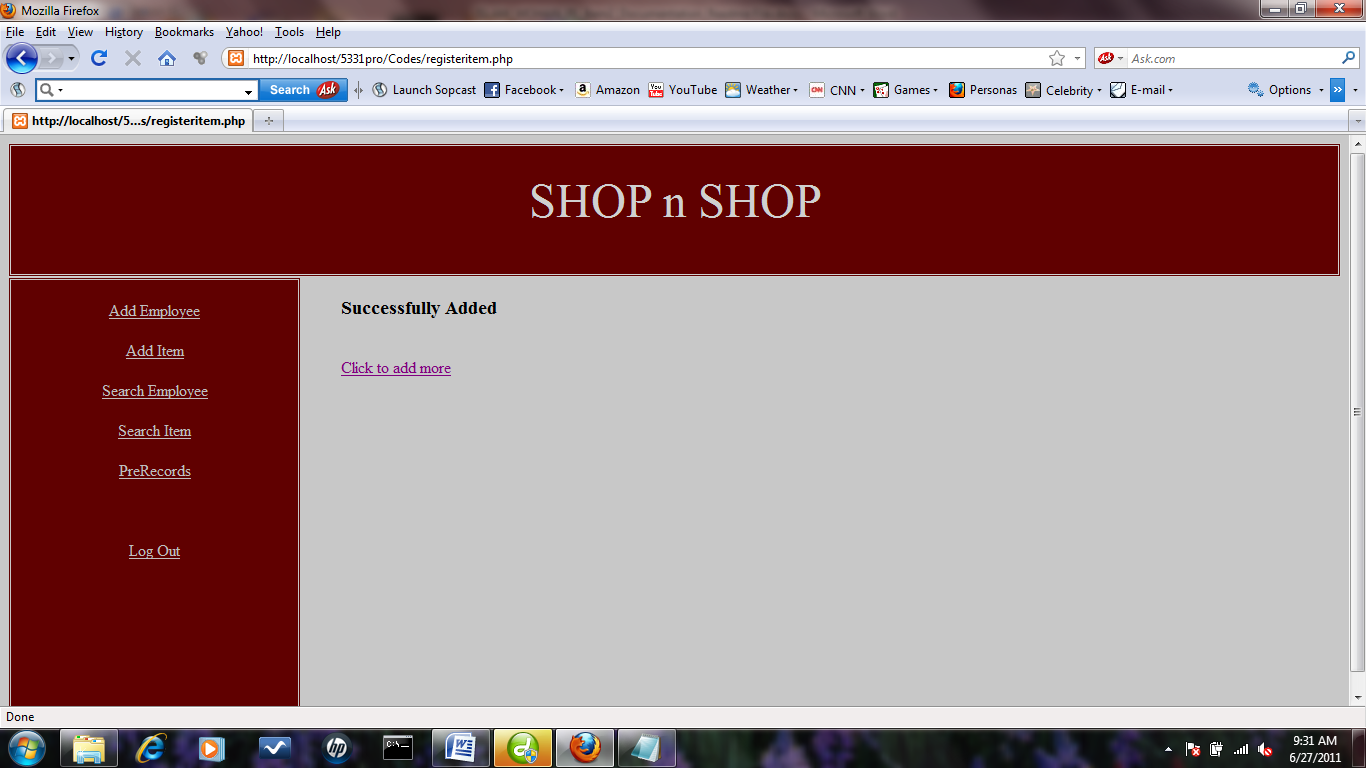


After clicking Submit, the message shows that Employee information has been inserted successfully. For wrong or already existing inputs, different message (unauthorized user) will be shown.

The next transaction is to add information about a new Item. We are adding the information and click submit, the following appears.



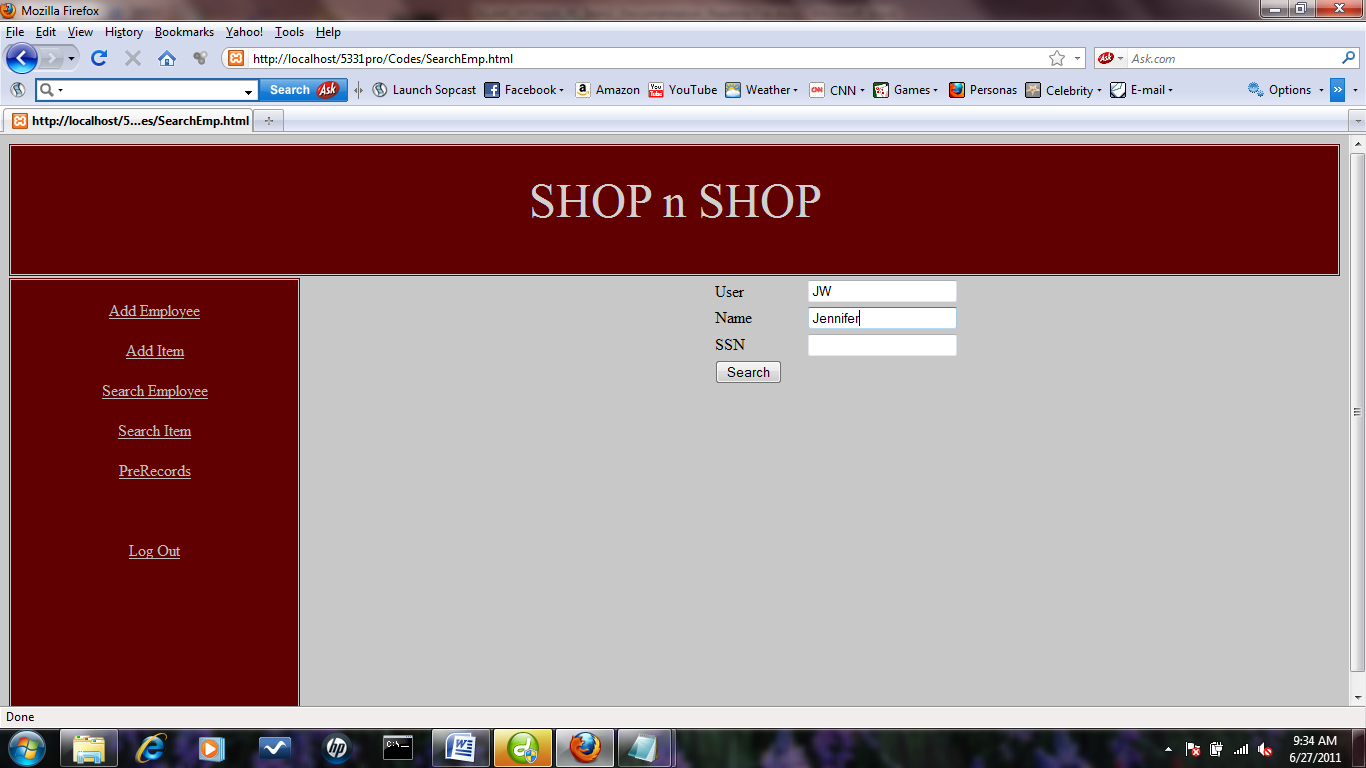
New Item information is inserted.



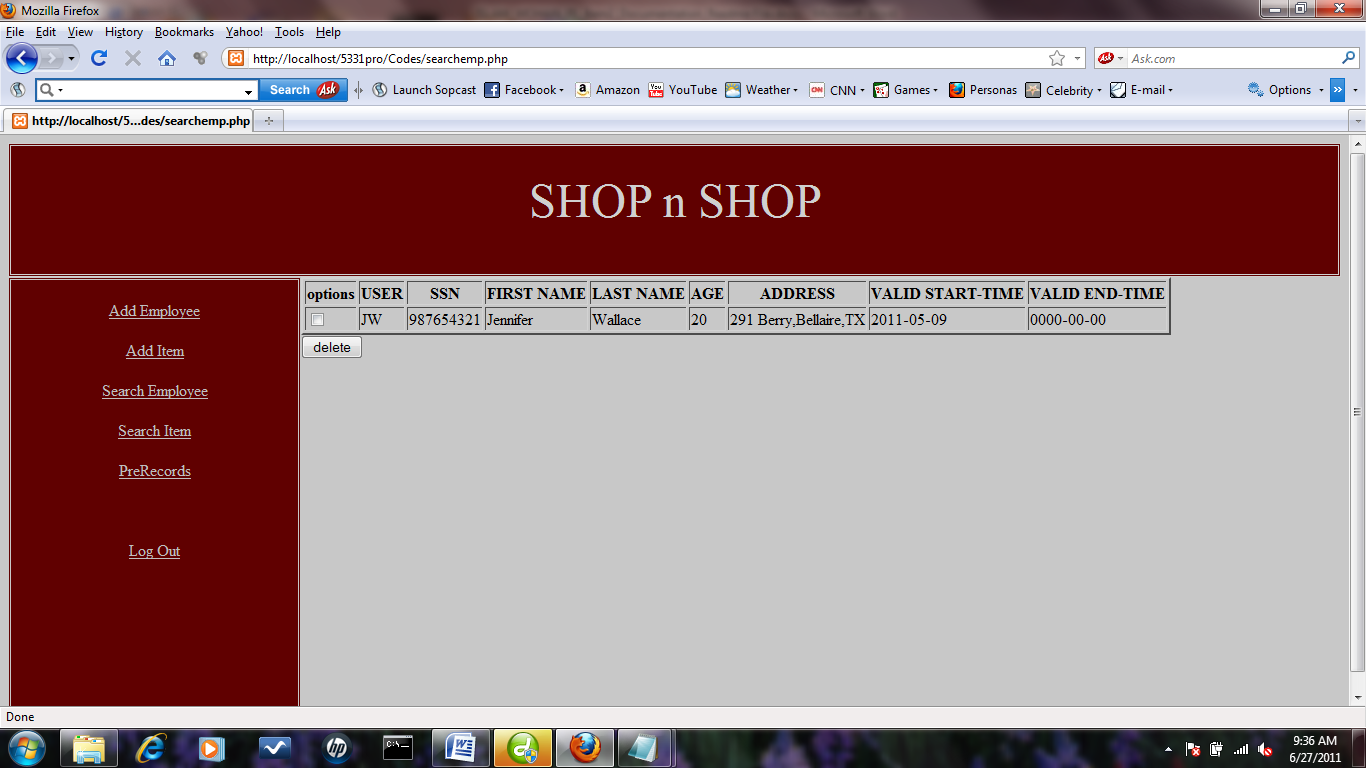
After clicking Submit, the message shows that Item information has been inserted successfully. For wrong or already existing inputs, different message (Product already exists please enter another name) will be shown.

Then Add Employee link opens up a search page that has complex search enabled to search employees in database and then the next page is used to delete the employee information. Employee can be searched by User, SSN and FName(searches on any field value even if any field is left blank or with wrong value, it searchs on rest of the correct value.

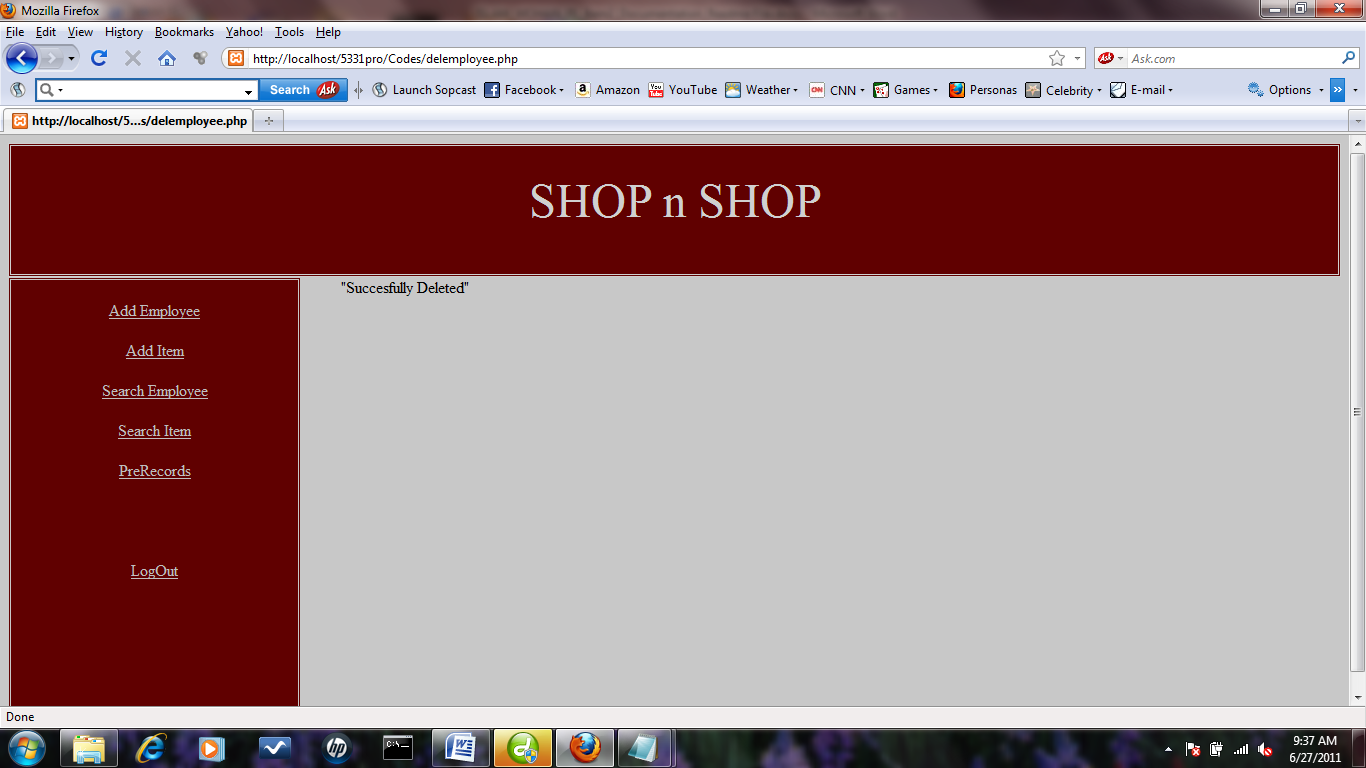
NOTE: atleast one field must be filled n must have correct value).



All three fields or any field with correct information will lead to employee information.



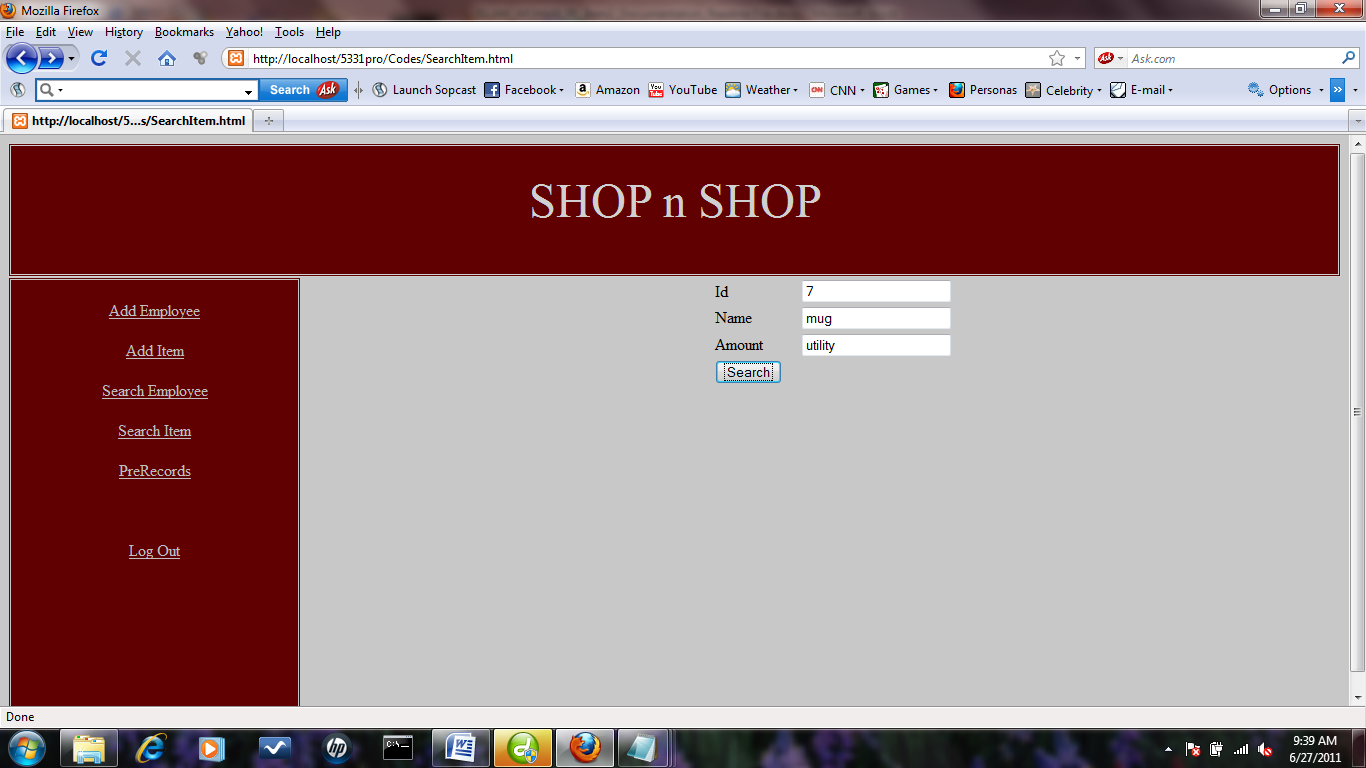
By selecting the tuple and then click delete will lead to deleting the Employee information.



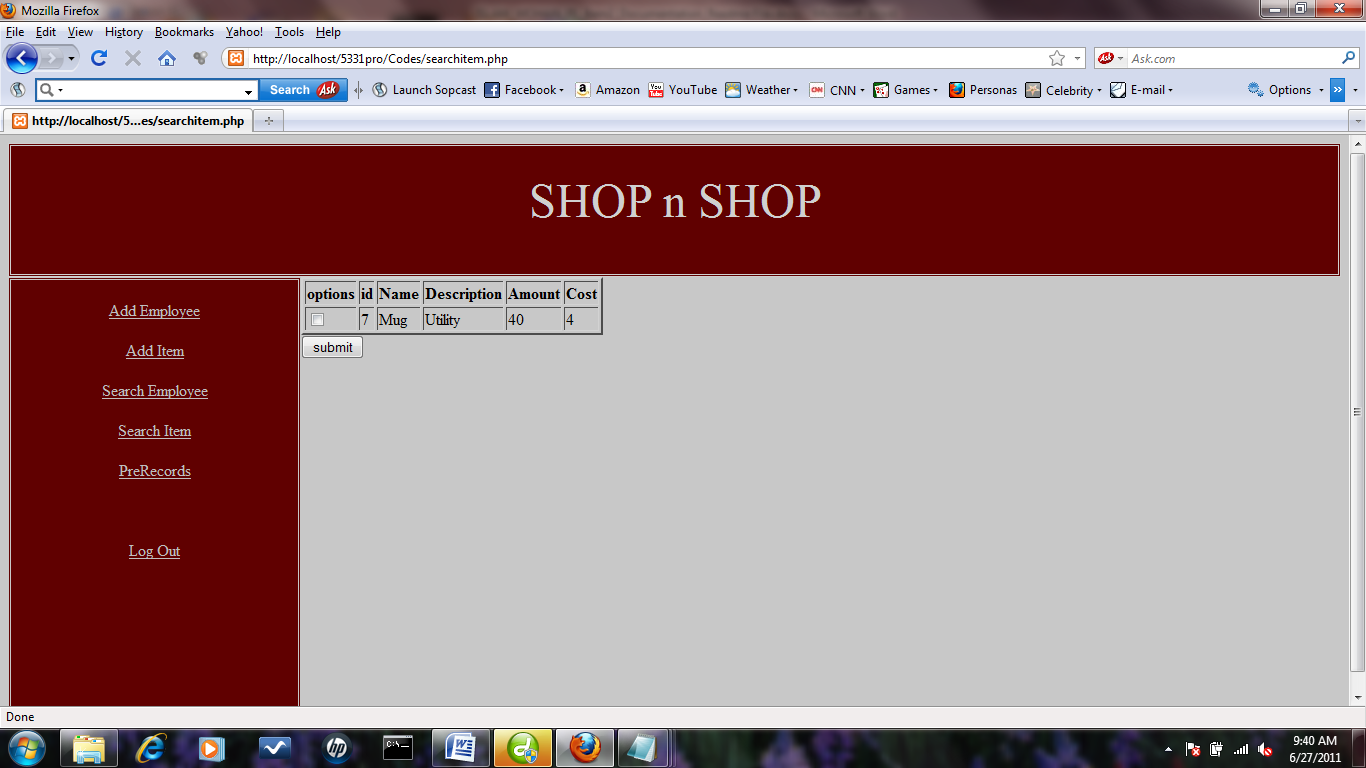
After clicking Submit, the message shows the information has been deleted successfully. Here prerecord trigger will update the vet of employee table as per the last time for the employee after the deletion.

Then Add item link opens up a search page that has complex search enabled to search items in database and then the next page is used to update the cost of the item. Item can be searched by Id, Name and Amount(searches on any field value even if any field is left blank or with wrong value, it searchs on rest of the correct value.

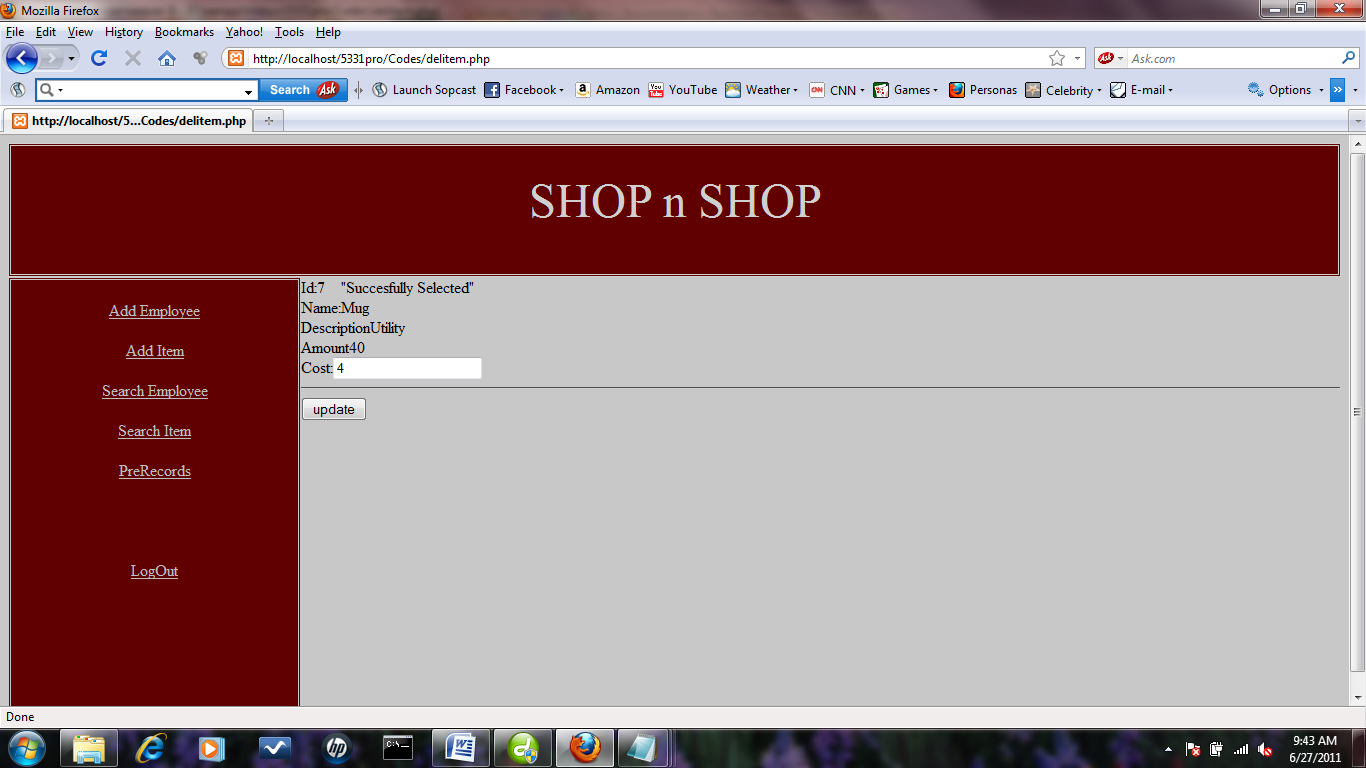
NOTE: atleast one field must be filled n must have correct value).



All three fields or any field with correct information will lead to item information.

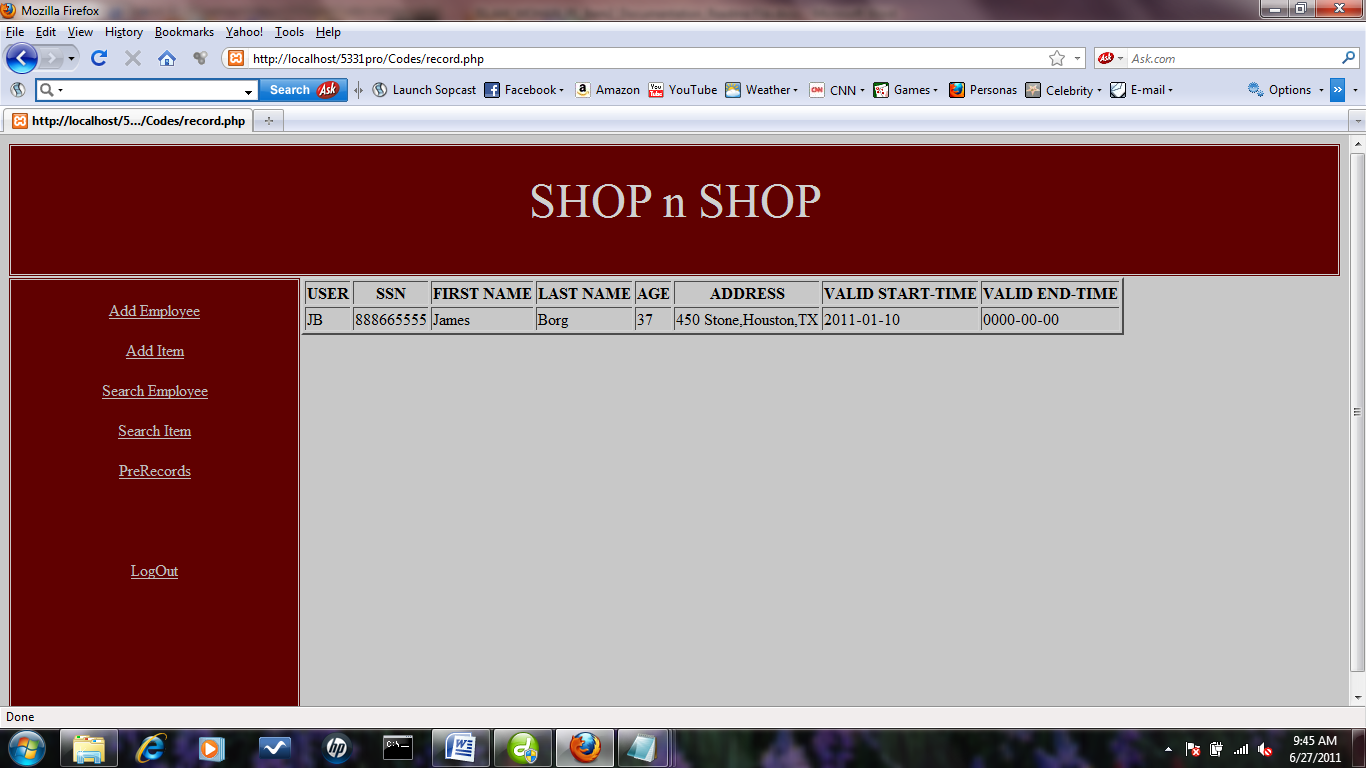


After clicking Submit, the message shows that information has been selected successfully.



Now can you update information by clicking the update button. Here a trigger icos will be executed to change information in the cost\_log and amount\_log based on the update of Item table.

Prerecords link will lead to complex search of an employee and then it show his database existence maintaining temporal information.



There will be logout link in every page and that will lead to login page after updating his lgout timing in the bonus table in database.

The above transactions are the required ones for the project; however the software contains few other links which are useful for some other purposes.