CHAPTER 8

Projects

This chapter presents several projects to be completed by the student using Java/JDBC and Oracle database or using PHP and MySQL database. These projects may be assigned as group projects with teams of two or three students. A written documentation as well as a class presentation of the project may be required.

8.1 STUDENT REGISTRATION System (GoLunar)

Consider the following relational database and sample data for the student registration database (written in Oracle SQL):

```
drop table students cascade constraints;
create table students (
  sid      number(4) primary key,
  password number(5),
  fname    varchar2(20),
  lname    varchar2(20),
  sType    varchar2(5) check (sType in ('GRAD','UGRAD')),
  major    char(4) check (major in ('CSC','MATH','POLS','HIST')),
  gradAssistant char(1) check (gradAssistant in ('Y','N')),
  inState char(1) check (inState in ('Y','N'))
);
insert into students values
  (1111,1111,'John','Davison','UGRAD','CSC','N','Y');
insert into students values
  (2222,2222,'Jacob','Oram','UGRAD','CSC','N','N');
insert into students values
  (3333,3333,'Ashish','Bagai','GRAD','CSC','Y','N');
insert into students values
  (4444,4444,'Joe','Harris','GRAD','CSC','N','Y');
insert into students values
  (5555,5555,'Andy','Blignaut','GRAD','CSC','N','Y');
insert into students values
  (6666,6666,'Pommie','Mbangwa','GRAD','CSC','N','Y');
insert into students values
  (7777,7777,'Ian','Healy','GRAD','CSC','N','Y');
insert into students values
  (8888,8888,'Dougie','Marillier','GRAD','CSC','N','Y');
--
drop table staff cascade constraints;
create table staff (
  tid  number(4) primary key,
  password number(5),
  fname varchar2(20),
  lname varchar2(20),
  staffType varchar2(10) check (staffType in ('REGISTRAR','DEPARTMENT'))
```

```
);
insert into staff values
  (1000,1000,'Venette','Rice','DEPARTMENT');
insert into staff values
  (2000,2000,'Alison','Payne','REGISTRAR');
--
create or replace view lunarUsers as
  (select sid uid, password, 'STUDENT' uType
   from   students) union
  (select tid uid, password, staffType uType
   from   staff);
--
drop table courses cascade constraints;
create table courses (
  cprefix  char(4),
  cno      number(4),
  ctitle   varchar2(50),
  chours   number(2),
  primary key (cprefix,cno)
);
insert into courses values ('CSC',1010,'Computers and Applications',3);
insert into courses values ('CSC',2010,'Introduction to Computer Science',3);
insert into courses values ('CSC',2310,'Intro to Programming in Java',3);
insert into courses values ('CSC',2311,'Introduction to Programming in C++',3);
insert into courses values ('CSC',3410,'Data Structures',3);
insert into courses values ('CSC',3210,'Computer Organization',3);
insert into courses values ('CSC',3320,'Systems Programming in Unix and C',3);
insert into courses values ('MATH',2211,'Calculus I',5);
insert into courses values ('MATH',2212,'Calculus II',5);
insert into courses values ('MATH',2420,'Discrete Mathematics',3);
insert into courses values ('CSC',6220,'Networks',4);
insert into courses values ('CSC',8220,'Advanced Networks',4);
insert into courses values ('CSC',6710,'Database',4);
insert into courses values ('CSC',8710,'Advanced Database',4);
insert into courses values ('CSC',6820,'Graphics',4);
insert into courses values ('CSC',8820,'Advanced Graphics',4);
insert into courses values ('POLS',1200,'Intro Political Sci',3);
--
drop table sections cascade constraints;
create table sections (
  term        char(2) check (term in ('FA','SP','SU')),
  year        number(4),
  crn         number(5),
  cprefix     char(4),
  cno         number(4),
  section     number(2),
  days        char(6),
  startTime   char(5), -- example 08.15, 13.30 etc.
  endTime     char(5),
  room        varchar2(10),
  cap         number(3),
  instructor  varchar2(30),
  auth        char(1) check (auth in ('Y','N')),
  primary key (term,year,crn),
  foreign key (cprefix,cno) references courses
);
--
```

```
insert into sections values
('SU',2002,10101,'CSC',1010,1,'MWF','09.00','09.50','105G',35,'Bhola','N');
insert into sections values
('SU',2002,10701,'POLS',1200,1,'TR','09.00','09.50','205Sp',25,'Jones','N');
--
insert into sections values
('FA',2002,10101,'CSC',2010,1,'MWF','09.00','09.50','105G',35,'Bhola','N');
insert into sections values
('FA',2002,10102,'CSC',2010,2,'MWF','10.00','10.50','105CS',40,'Henry','N');
insert into sections values
('FA',2002,10103,'CSC',2310,1,'MWF','12.00','12.50','106G',30,'Henry','N');
insert into sections values
('FA',2002,10104,'CSC',2311,1,'MWF','15.00','15.50','205G',35,'Liu','N');
insert into sections values
('FA',2002,10201,'CSC',6220,1,'TR','19.00','20.40','405G',25,'Hundewale','N');
insert into sections values
('FA',2002,10202,'CSC',6710,1,'TR','16.00','17.15','115CS',25,'Madiraju','N');
insert into sections values
('FA',2002,10203,'CSC',8820,1,'MWF','09.00','09.50','605G',25,'Owen','N');
insert into sections values
('FA',2002,10301,'MATH',2211,1,'TR','11.00','12.50','305G',35,'Li','N');
insert into sections values
('FA',2002,10302,'MATH',2211,2,'MWF','09.00','10.50','106GB',35,'Davis','N');
--
--This data will be loaded into the database in your application program
--insert into sections values
--('SP',2003,10101,'CSC',2010,1,'MWF','09.00','09.50','105G',35,'Bhola','N');
--insert into sections values
--('SP',2003,10102,'CSC',2010,2,'MWF','10.00','10.50','105CS',40,'Henry','N');
--insert into sections values
--('SP',2003,10103,'CSC',2310,1,'MWF','12.00','12.50','106G',30,'Henry','N');
--insert into sections values
--('SP',2003,10104,'CSC',2311,1,'MWF','15.00','15.50','205G',35,'Liu','N');
--insert into sections values
--('SP',2003,10201,'CSC',6220,1,'TR','19.00','20.40','405G',25,'Hundewale','N');
--insert into sections values
--('SP',2003,10202,'CSC',6710,1,'TR','16.00','17.15','115CS',25,'Madiraju','N');
--insert into sections values
--('SP',2003,10203,'CSC',8220,1,'MWF','09.00','09.50','605G',25,'Bourgeois','Y');
--insert into sections values
--('SP',2003,10301,'MATH',2211,1,'TR','11.00','12.50','305G',35,'Li','N');
--insert into sections values
--('SP',2003,10302,'MATH',2211,2,'MWF','09.00','10.50','606GB',35,'Miller','N');
--insert into sections values
--('SP',2003,10303,'MATH',2212,1,'MWF','09.00','10.50','706GB',35,'Davis','N');
--insert into sections values
--('SP',2003,10304,'MATH',2420,1,'TR','14.00','14.50','106GB',35,'Domke','N');
--insert into sections values
--('SP',2003,10405,'CSC',8710,1,'MW','17.30','18.45','206GB',35,'Dogdu','N');
--insert into sections values
--('SP',2003,10406,'CSC',8820,1,'TR','19.15','20.55','306GB',3,'Owen','N');
--
drop table enrolls cascade constraints;
create table enrolls (
  sid    number(4),
  term   char(2) check (term in ('FA','SP','SU')),
  year   number(4),
  crn    number(5),
  grade  char(2) check (grade in ('A','B','C','D','F','I','IP','S','U')),
  primary key (sid,term,year,crn),
```

```
  foreign key (sid) references students,
  foreign key (term,year,crn) references sections
);
--
insert into enrolls values (1111,'SU',2002,10101,'A');
insert into enrolls values (1111,'SU',2002,10701,'C');
--
insert into enrolls values (1111,'FA',2002,10101,null);
insert into enrolls values (1111,'FA',2002,10103,null);
insert into enrolls values (1111,'FA',2002,10301,null);
insert into enrolls values (3333,'FA',2002,10201,null);
insert into enrolls values (3333,'FA',2002,10202,null);
insert into enrolls values (3333,'FA',2002,10203,null);
--
drop table authorizations cascade constraints;
create table authorizations (
  term   char(2) check (term in ('FA','SP','SU')),
  year   number(4),
  crn    number(5),
  sid    number(4),
  authType char(4) check (authType in ('OVFL','AUTH')),
  primary key (term,year,crn,sid,authType),
  foreign key (sid) references students,
  foreign key (term,year,crn) references sections
);
--
drop table fixedFee cascade constraints;
create table fixedFee (
  feeName varchar2(30) primary key,
  fee number(5,2)
);
--
insert into fixedFee values ('Technology Fee',75.00);
insert into fixedFee values ('Health Fee',30.00);
insert into fixedFee values ('Activity Fee',65.00);
insert into fixedFee values ('Transportation Fee',25.00);
--
drop table variableFeeRate cascade constraints;
create table variableFeeRate (
  sType varchar2(6)
        check (sType in ('GRAD','UGRAD')),
  inOrOutOfState varchar2(10)
        check (inOrOutOfState in ('INSTATE','OUTOFSTATE')),
  fee number(6,2),
  primary key (sType,inOrOutOfState)
);
--
insert into variableFeeRate values ('GRAD','INSTATE',125.00);
insert into variableFeeRate values ('GRAD','OUTOFSTATE',500.00);
insert into variableFeeRate values ('UGRAD','INSTATE',100.00);
insert into variableFeeRate values ('UGRAD','OUTOFSTATE',400.00);
```

The database consists of the following tables:

1. `Students`: This table records information about students. The `gradAssistant` attribute records whether the student is a graduate assistant or not. The graduate assistants

automatically qualify for a full tuition waiver (they still have to pay the fixed fee). The `instate` attribute records whether the student is an in-state student or not. Again, this has an impact on the fees the student would pay.

2. `Staff`: This table records information about staff users of the system. There are two categories of staff: "`Registrar`" and "`Department`". These users would have different capabilities and functions in the application to be developed.

   Note: A view called `lunarUsers` is created to provide a simple way to authenticate users of the system.

3. `Courses`: This table records information about courses in the university catalog which includes course number, title and credit hours. The credit hours value will be used in calculating the GPA in the student's transcript.
4. `Sections`: This table records the course offerings for each term and includes the term, year, and course record number (`crn`), a unique number assigned to course offerings for a specific term and year. The table also includes start time and end time and meeting days as well as the name of the instructor. Finally, this table records a boolean value (yes or no) called `auth` to indicate if the registration for this course is open to all or is done only by authorization.
5. `Enrolls`: This table records information about which student has registered for which course offering.
6. `Authorizations`: This table records authorizations given to students for specific course offerings. Two types of authorizations are given: `OVFL` for overflow, i.e. allows students to register in a course offering that does not have any open seats, and `AUTH` for authorization to register in a course offering that is designated as a authorization only course offering.
7. `FixedFee`: This table records information about all fixed fees a student is required to pay each term they register.
8. `VariableFeeRate`: This table records per credit hour fee rate for different categories of students (graduate vs undergraduate students and in-state vs out-of-state students).

You will implement a University Registration System in Java using JDBC.

There are 3 kinds of database users:

1. **Registrar Staff:** These users will have the ability to load the database tables, make changes to courses, sections, fee details etc.
2. **Department Staff:** These users will have the ability to authorize students into sections, overflow students into sections, add assistantship information to the system, generate class lists etc.
3. **Student:** These users will be able to register for classes, see their schedules, see fee detail, see transcripts etc.

The following real-world constraints need to be enforced by your Java program:

1. Undergraduate students are not allowed to register for graduate courses numbered 6000 and above.
2. Students should not be allowed to register for a class which is FULL unless they have an overflow.
3. Students should not be allowed to register for a class which is listed as AUTHORIZATION ONLY unless they have an authorization.
4. Undergraduate students are not allowed to register for more than 20 hours in a semester and the limit for graduate students is 15.
5. Students cannot register for two classes that overlap in meeting time.

The Java application will be a terminal-based program that has the following interactions with the users. Based on the username, the program should determine the type of user and provide the appropriate menu.

Department Staff Menu:

```
$ java GoLunar OracleId
Oracle Password:xxxxxx
Semester (e.g. FA2003,SP2003,SU2003): SP2003
Username: 1000
Password:

*********************************************************************
***                                                               ***
***    Welcome to the GoLunar - Online Registration System        ***
***        Venette Rice - Department Staff                        ***
***                                                               ***
*********************************************************************

                         1. Authorize Student into Section

                         2. Overflow Student into Section

                         3. Add Assistantship on System

                         4. Generate Class List

                         q. Quit

Type in your option:
```

Option 1 Interface:

```
CRN:10101
SID:1111
Student John Davison authorized into CRN 10101, CSC 2010.
OR
No need to authorize - This section does not need authorization.
```

Option 2 Interface:

```
CRN:10101
SID:1111
No need to overflow - Space still available in this section.
OR
Student John Davison overflowed into CRN 10101, CSC 2010.
```

Option 3 Interface:

```
Student Id: 3333
Ashish Bagai (3333) has been added to the Assistantship List.
```

Option 4 Interface:

```
CRN: 10101

CSC 2010, Introduction to Computer Science
SP 2003
Instructor: Bhola

SID           LNAME           FNAME
-------------------------------------------
1111          Davison         John
2222          Oram            Jacob
...
...
```

Student Menu:

```
$ java GoLunar OracleId
Oracle Password:xxxxxx
Semester (e.g. FA2003,SP2003,SU2003): SP2003
Username: 1111
Password:
*********************************************************************
***                                                               ***
***    Welcome to the GoLunar - Online Registration System        ***
***        John Davison - Student                                 ***
***                                                               ***
*********************************************************************

                          1. Add a Section

                          2. Drop a Section

                          3. See Schedule for a Term

                          4. See Fee detail

                          5. See Transcript

                          q. Quit

Type in your option: 1
```

Option 1 Interface:

```
CRN: 10101
CSC2010, Introduction to Computer Science ADDED.
OR
Appropriate Error Message.
```

Option 2 Interface:

```
CRN: 10101
CSC2010, Introduction to Computer Science DROPPED.
OR
Appropriate Error Message.
```

Option 3 Interface:

```
Term: FA2002

CRN   Course  Title                                Days Time        Room Instructor
-------------------------------------------------------------------------------
10101 CSC2010 Introduction to Computer Science MWF  09.00-09.50 105G Bhola
...
...
```

Option 4 Interface:

```
Term: sp2003

Spring 2003

  Tuition - InState
      (12 hours)     1,500.00
  Technology Fee        75.00
  Health Fee            30.00
  Activity Fee          65.00
  Transportation Fee    25.00
                     --------
                     1,695.00
                     --------
```

Option 5 Interface:

```
Summer 2002
  CSC  1010 10101 Computers and Applications          3 A 12.00
  POLS 1200 10701 Intro Political Sci                 3 C  6.00
       Semester GPA: 3.00   GPA: 3.00

Fall 2002
  CSC  2010 10101 Introduction to Computer Science    3 A 12.00
  CSc  2310 Introduction to Programming in Java       3 A 12.00
  Math 2211 Calculus I                                5 B 15.00
       Semester GPA: 3.54    GPA: 3.35
...
...
```

Registrar Staff Menu:

```
$ java GoLunar OracleId
Oracle Password:xxxxxx
Semester (e.g. FA2003,SP2003,SU2003): SP2003
Username: 2000
Password:

*********************************************************************
***                                                               ***
***    Welcome to the GoLunar - Online Registration System        ***
***        Alison Payne - Registrar Staff                         ***
***                                                               ***
*********************************************************************

                        1. Load Sections from File

                        2. Load Grades from File

                        3. Increase Section Cap

                        4. Display Term Schedule

                        5. Display Student Transcript

                        6. Display Student Schedule and Fee Detail

                        q. Quit


Type in your option:
```

Option 1 Interface:

```
File Name: sections.dat
Sections Loaded
```

Option 2 Interface:

```
File Name: grades.dat
Grades Loaded
```

Option 3 Interface:

```
CRN: 10101
Old Capacity is 35
New Capacity: 45
Cap Updated for CRN 10101.
```

Option 4 Interface:

```
CRN   Course  Sec  Days  Time        Room  Cap Cur Avail Instructor Auth
-------------------------------------------------------------------------
10101 CSC2010 1    MWF   09.00-09.50 105G  35   3   32   Bhola      N
10102 CSC2010 2    MWF   10.00-10.50 105CS 40   5   35   Henry      N
...
```

```
...
```

Option 5 is similar to student option except here the system should accept student id as input and display that student's transcript.

Option 6 is similar to student options except here the system should accept student id (in addition to term) as input and display that student's term schedule and fee detail for the particular term.

Sample files for loading data in the Registrar's options are available in

sections.dat

```
SP
2003
10101,CSC,2010,1,MWF,09.00,09.50,105G,35,Bhola,N
10102,CSC,2010,2,MWF,10.00,10.50,105CS,40,Henry,N
10103,CSC,2310,1,MWF,12.00,12.50,106G,30,Henry,N
10104,CSC,2311,1,MWF,15.00,15.50,205G,35,Liu,N
10201,CSC,6220,1,TR,19.00,20.40,405G,25,Hundewale,N
10202,CSC,6710,1,TR,16.00,17.15,115CS,25,Madiraju,N
10203,CSC,8220,1,MWF,09.00,09.50,605G,25,Bourgeois,Y
10301,MATH,2211,1,TR,11.00,12.50,305G,35,Li,N
10302,MATH,2211,2,MWF,09.00,10.50,606GB,35,Miller,N
10303,MATH,2212,1,MWF,09.00,10.50,706GB,35,Davis,N
10304,MATH,2420,1,TR,14.00,14.50,106GB,35,Domke,N
10405,CSC,8710,1,MW,17.30,18.45,206GB,35,Dogdu,N
10406,CSC,8820,1,TR,19.15,20.55,306GB,3,Owen,N
```

grades.dat

```
FA
2002
1111,10101,A
1111,10103,A
1111,10301,B
3333,10201,B
3333,10202,B
3333,10203,A
```

8.2 Online Book Store Database System

Consider the following relational database schema written in Oracle SQL for an online book store application:

```
drop table books cascade constraints;
create table books (
  isbn char(10),
  author varchar2(100) not null,
  title varchar2(128) not null,
  price number(7,2) not null,
  subject varchar2(30) not null,
  primary key (isbn)
);
```

```
drop table members cascade constraints;
create table members (
  fname varchar2(20) not null,
  lname varchar2(20) not null,
  address varchar2(50) not null,
  city varchar2(30) not null,
  state varchar2(20) not null,
  zip number(5) not null,
  phone varchar2(12),
  email varchar2(40),
  userid varchar2(20),
  password varchar2(20),
  creditcardtype varchar2(10)
    check(creditcardtype in ('amex','discover','mc','visa')),
  creditcardnumber char(16),
  primary key (userid)
);
drop table orders cascade constraints;
create table orders (
  userid varchar2(20) not null,
  ono number(5),
  received date not null,
  shipped date,
  shipAddress varchar2(50),
  shipCity varchar2(30),
  shipState varchar2(20),
  shipZip number(5),
  primary key (ono),
  foreign key (userid) references members
);
drop table odetails cascade constraints;
create table odetails (
  ono number(5),
  isbn char(10),
  qty number(5) not null,
  price number(7,2) not null,
  primary key (ono,isbn),
  foreign key (ono) references orders,
  foreign key (isbn) references books
);
drop table cart cascade constraints;
create table cart (
  userid varchar2(20),
  isbn char(10),
  qty number(5) not null,
  primary key (userid,isbn),
  foreign key (userid) references members,
  foreign key (isbn) references books
);
```

The database consists of five tables:

1. `Books`: This table records information about the books on sale in the book store. Each book is classified under a “subject” to enable subject searches.

2. `Members`: This table records information about members of the application. Each member chooses their own user id and password at the time of registration.
3. `Orders`: This table records information about orders placed by members place orders. The orders may contain one or more books and the details of the order are kept in a separate table. A unique order number is generated by the system.
4. `OrderDetails`: This table records information about each order including the isbn and quantity of books in the order.
5. `Cart`: This table contains isbn and quantity of each book placed in the shopping cart of a member. Once a member checks out, the shopping cart is emptied and an order is created.

The book store application should be developed as a terminal application in Java and should be implemented in three phases:

Phase I of the project requires:

(a) Each student to create data for approximately 10 books for two different subjects (the subjects may be assigned by the instructor of the class to each student). The instructor may then consolidate the data into a large data set and give it out to the entire class. This is an easy way to create a large data set of books.

(b) Each student to build program the following interface implementing only the member registration and member login functions:

```
$ java OnlineBookStore

**********************************************************************
***                                                                ***
***             Welcome to the Online Book Store                   ***
***                                                                ***
**********************************************************************
                          1. Member Login

                          2. New Member Registration

                          q. Quit

Type in your option: 2

Welcome to the Online Book Store
    New Member Registration

Enter first name: Raj
Enter last name: Sunderraman
Enter street address: 123 Main Street
Enter city: Atlanta
Enter state: GA
Enter zip: 30303
Enter phone: 555-1212
Enter email address: raj@cs.gsu.edu
Enter userID: raj
Enter password: raj
```

```
Do you wish to store credit card information(y/n): y
Enter type of Credit Card(amex/visa): amex
Enter Credit Card Number: 121212121212121
Invalid Entry
Enter Credit Card Number: 1212121212121212
Invalid Entry
Enter Credit Card Number: 12121212121212

You have registered successfully.
Name:                Raj Sunderraman
Address:             123 Main Street
City:                Atlanta GA 30303
Phone:               555-1212
Email:               raj@cs.gsu.edu
UserID:              raj
Password:            raj
CreditCard Type:     amex
CreditCard Number:   12121212121212
Press Enter to go back to Menu

**************************************************************************
***                                                                    ***
***             Welcome to the Online Book Store                       ***
***                                                                    ***
**************************************************************************
                         1. Member Login

                         2. New Member Registration

                         q. Quit
Type in your option: 1

Enter userID: raj
Enter password: raj
**************************************************************************
***                                                                    ***
***                          Welcome to Online Book Store              ***
***                               Member Menu                          ***
***                                                                    ***
**************************************************************************
                     1. Browse by Subject

                     2. Search by Author/Title/Subject

                     3. View/Edit Shopping Cart

                     4. Check Order Status

                     5. Check Out

                     6. One Click Check Out

                     7. View/Edit Personal Information

                     8. Logout
Type in your option: 8
You have successfully logged out.
```

Phase II of the project requires the student to implement the following member functions:

1. Browse by Subject: This option should first list all subjects alphabetically; It then allows user to choose one subject; Upon choosing a subject, the program displays book details (2 books at a time on a screen); The option allows user to
(a) enter isbn to put in cart;
(b) press `ENTER` to return to main menu
(c) press `n ENTER` to continue browsing

User Interface follows:

```
Type in your option: 1

1. Cooking
2. Jokes
3. Sports

Enter your choice: 3

5 books available on this Subject

Author: Dom Parker
Title:  1,001 Baseball Questions Your Friends Can't Answer
ISBN:   0451191323
Price:   22.46
Subject Sports

Author: Timothy Jacobs
Title:  100 Atheletes Who Shaped Sports History
ISBN:   0912517131
Price:   32.56
Subject Sports


Enter ISBN to add to Cart or
n Enter to browse or
ENTER to go back to menu:
0451191323
Enter quantity: 2


Author: Michael Dregni
Title:  100 Years of Fishing
ISBN:   0896584305
Price:   15.95
Subject Sports

Author: David Claerbaut
Title:  The 1999 NBA Analyst: The Science of Hoops Magic
ISBN:   0878332103
Price:   20.95
Subject Sports
```

```
Enter ISBN to add to Cart or
n Enter to browse or
ENTER to go back to menu: n

Author: Sports Collectors Digest
Title:  1999 Sports Collectors Almanac (Serial)
ISBN:   0987654234
Price:   17.56
Subject Sports

Enter ISBN to add to Cart or
n Enter to browse or
ENTER to go back to menu:
0987654234
Enter quantity: 1
```

6. One Click Check Out

This option should move items in the cart to the `order` and `odetails` tables. Cart is emptied in the process and an invoice is printed. Shipping address is same as member address in this option. User Interface follows:

```
                    Invoice for Order no.117

     Shipping Address                    Billing address
     Name:     Raj Sunderraman           Name:     Raj Sunderraman
     Address:  123 Main Street           Address:  123 Main Street
               Atlanta                             Atlanta
               GA 33333                            GA 33333

-------------------------------------------------------------------------------
ISBN        Title                                              $  Qty   Total
-------------------------------------------------------------------------------
0451191323  1,001 Baseball Questions Your Friends Can't Answer 22.45  1   22.45
0987654234  1999 Sports Collectors Almanac(Serial)             17.55  1   17.55
-------------------------------------------------------------------------------
Total =                                                                  $40.01
-------------------------------------------------------------------------------

Press enter to go back to Menu
```

4. Check Order Status

This option should list all orders for member and should allow user to choose one order to see details. User Interface follows:

```
Orders placed by Raj Sunderraman
-----------------------------------------------------------
ORDER NO    RECEIVED DATE                SHIPPED DATE
-----------------------------------------------------------
117         3-1-2001                     3-3-2001
-----------------------------------------------------------

Enter the Order No to display its details or (q) to quit: 117
```

```
                    Details for Order no.117

     Shipping Address                    Billing address
     Name:     Raj Sunderraman           Name:     Raj Sunderraman
     Address:  123 Main Street           Address:  123 Main Street
               Atlanta                             Atlanta
               GA 33333                            GA 33333

-------------------------------------------------------------------------------
ISBN        Title                                              $  Qty   Total
-------------------------------------------------------------------------------
0451191323  1,001 Baseball Questions Your Friends Can't Answer 22.45  1   22.45
0987654234  1999 Sports Collectors Almanac(Serial)             17.55  1   17.55
-------------------------------------------------------------------------------
Total =                                                                 $40.01

Press Enter to go back to Menu
```

Phase III of the project requires the student to implement the following member functions:

2. Search by Author/Title

This option should provide 3 sub-options:

1. Author Search
2. Title Search
3. Go Back to Member Menu

In the Author or Title search sub-option, the user may enter a substring and the system should respond with all books which contain the substring in the title/author. The display should be done 2 books at a time on a screen.

The system should also allow user to enter isbn to put in cart;

to press `ENTER` to return to main menu

to press `n ENTER` to continue browsing

User Interface follows:

```
1. Author Search
2. Title Search
3. Go Back to Member Menu

Type in your option: 2

Enter title or part of the title: cook
2 books found


Author: Irma S. Rambauer
Title:  Joy of Cooking
ISBN:   0452279232
Price:   15.25
Subject Cooking


Author: Jennifer E. Darling
Title:  Better Homes and Gardens New Cook Book
```

```
ISBN:   0696201887
Price:   21.96
Subject Cooking


Enter ISBN to add to Cart or
      Enter to browse or
      n ENTER to return to menu: 0696201887
Enter quantity: 1


1. Author Search
2. Title Search
3. Go Back to Member Menu

Type in your option: 2

Enter title or part of the title: Computer
0 books found

Enter ISBN to add to Cart or
      Enter to browse or
      n ENTER to return to menu: n

1. Author Search
2. Title Search
3. Go Back to Member Menu

Type in your option: 1

Enter name or part of the name: am
1 books found

Author: Irma S. Rambauer
Title:  Joy of Cooking
ISBN:   0452279232
Price:   15.25
Subject Cooking

Enter ISBN to add to Cart or
      Enter to browse or
      n ENTER to return to menu: 0452279232
Enter quantity: 2


1. Author Search
2. Title Search
3. Go Back to Member Menu

Type in your option: 3
```

3. View/Edit Shopping Cart

This option should show the contents of the cart; It should then provide options to delete items or edit (change quantity) items. User Interface (for delete and update cart) follows:

```
Current Cart Contents:

ISBN        Title                                                    $  Qty   Total
-------------------------------------------------------------------------------
0696201887  Better Homes and Gardens New Cook Book             21.95  1   21.95
0452279232  Joy of Cooking                                     15.25  2   30.50
-------------------------------------------------------------------------------
Total =                                                                  $52.45

Enter d to delete item
e to edit cart or
q to go back to Menu: d
Enter isbn of item: 0452279232
Delete Item Completed
Press enter to go back to Menu

Type in your option: 3

Current Cart Contents:

ISBN        Title                                                    $  Qty   Total
-------------------------------------------------------------------------------
0696201887  Better Homes and Gardens New Cook Book             21.95  1   21.95
-------------------------------------------------------------------------------
Total =                                                                  $21.95

Enter d to delete item
e to edit cart or
q to go back to Menu: e
Enter isbn of item: 0696201887
Enter new Quantity: 2
Edit Item Completed
Press enter to go back to Menu
```

5. Check Out

This option should display invoice; request user of they want to provide shipping address (if no use current address in file for shipping); Also this option should ask if a new credit card should be used. Finally, an invoice should be printed. User Interface follows:

```
Current Cart Contents:

ISBN        Title                                                    $ Qty   Total
-------------------------------------------------------------------------------
0696201887  Better Homes and Gardens New Cook Book             21.95  2   43.91
-------------------------------------------------------------------------------
Total                                                                    $43.91
-------------------------------------------------------------------------------

Proceed to check out(Y/N): y
Do you want to enter new shipping address(y/n): y
Enter first name: John
Enter last name: Smith
```

```
Enter street: 123 Elm Street
Enter city: Atlanta
Enter state: GA
Enter zip: 11111
Do you want to enter new CreditCard Number(y/n): n

                    Invoice for Order no.118

     Shipping Address                    Billing address
     Name:     John Smith                Name:     Raj Sunderraman
     Address:  123 Elm Street            Address:  123 Main Street
               Atlanta                             Atlanta
               GA 11111                            GA 33333

-------------------------------------------------------------------------------
ISBN        Title                                          $  Qty   Total
-------------------------------------------------------------------------------
0696201887  Better Homes and Gardens New Cook Book      21.95  2   43.91
-------------------------------------------------------------------------------
Total =                                                          $43.91
-------------------------------------------------------------------------------

Press enter to go back to Menu
```

8.3 Online Shopping System

Using PhP and MySQL, implement a Web-based application for an online video-store. The online video-store maintains an inventory of DVDs. Customers become member of this online store. They are able to search for DVDs of their interest and add DVDs to their shopping cart. At any time, they are able to edit the shopping cart and also are able to check out. The initial login screen shown in the Figure 8.1 allows an existing customer to sign in or a new customer to register. The new customer registration screen is shown in Figure 8.2. Upon successful sign-in, the 3-frame page shown in Figure 8.3 is displayed. As one can see, there are six different options for the customer:

1. **Search by Keyword:** This option allows the customer to perform a keyword search of the DVD titles (substring; case insensitive comparison). Successful matches are shown on the right frame (Figure 8.4). The customer may then choose certain quantities of the DVDs and add them to the shopping cart. Upon successful addition to the shopping cart, a message should be shown on the right frame.
2. **View/Edit Cart:** Upon clicking this option, the system should display the shopping cart on the right side frame (Figure 8.5). Here, the customer may edit the shopping cart by changing quantities including replacing a value with a zero. Upon submission, the cart should be updated and a message should be displayed.
3. **Update Profile:** This option brings up the customers profile (Figure 8.6) on the right frame. The user may modify any field and submit. Upon successful update, a message should be displayed.
4. **Check Order Status:** This option allows the customer to see all their orders (Figure 8.7). Upon clicking the order number link the details for that order should be displayed in a tabular format.

5. **Check Out:** Upon clicking this link, the system should empty the cart and move the items into the `orders` and `odetails` tables. An invoice (Figure 8.8) should be printed on the right frame.
6. **Logout:** Upon logout, the system should display 3 Options (Figure 8.9) to the user. The user may check out, save cart and logout or empty cart and logout. Upon checkout a similar action should take place as earlier. Upon the other two options, appropriate action should take place and a message should be displayed. If the cart was empty to begin with these 3 options should not be shown and the customer should be logged out.

The database schema for the online shopping cart example is shown below:

```
create table parts(
  pno      integer(5) not null,
  pname    varchar(30),
  qoh      integer,
  price    decimal(6,2),
  olevel   integer,
  primary key (pno));

create table customers (
  cno      integer(10) not null auto_increment=100,
  cname    varchar(30),
  street   varchar(50),
  city     varchar(30),
  state    varchar(30),
  zip      integer(5),
  phone    char(12),
  email    varchar(50),
  password varchar(15),
  primary key (cno));

create table cart(
  cartno   integer(10) not null auto_increment,
  cno      integer(10) not null,
  pno      integer(5) not null,
  qty      integer,
  primary key (cartno, pno),
  foreign key (cno) references customers,
  foreign key (pno) references parts);

create table orders (
  ono      integer(5) not null auto_increment=1000,
  cno      integer(10),
  received date,
  shipped  date,
  primary key (ono),
  foreign key (cno) references customers);

create table odetails (
  ono      integer(5) not null,
  pno      integer(5) not null,
  qty      integer,
  primary key (ono,pno),
  foreign key (ono) references orders,
```

```
foreign key (pno) references parts);
```

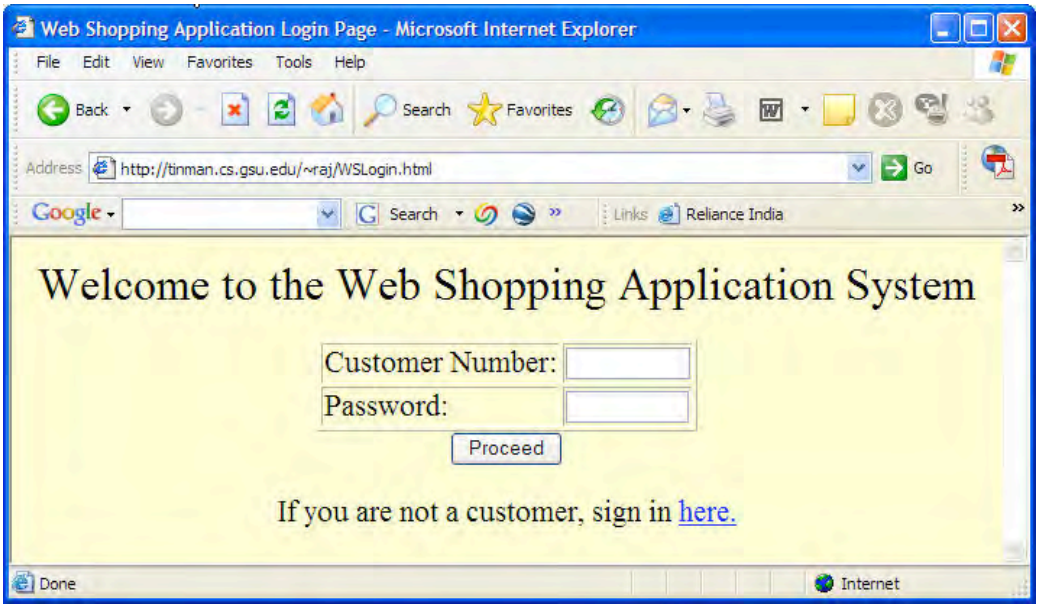

Figure 8.1 Web Shopping – Initial Screen

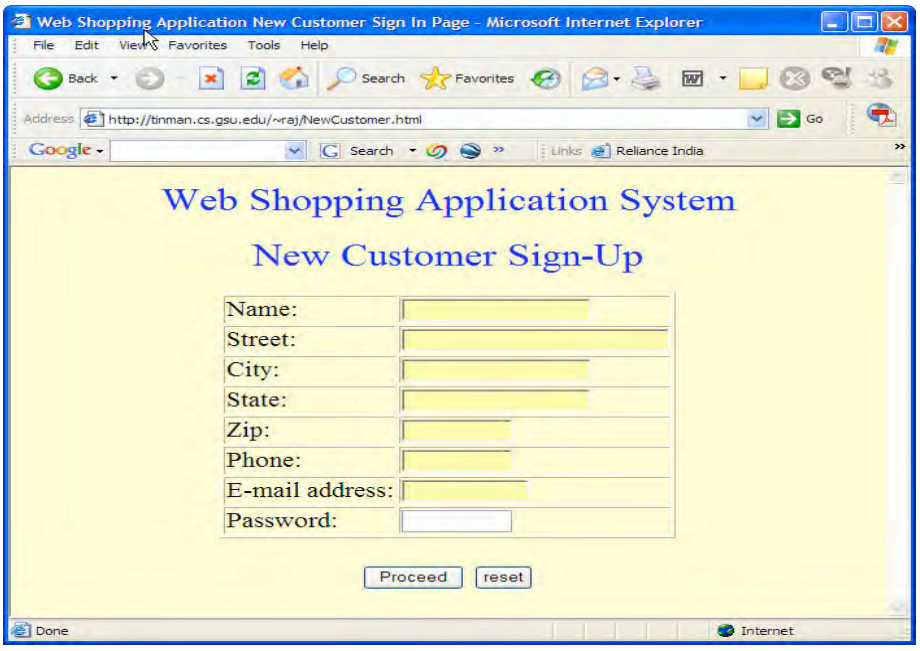

Figure 8.2 Web Shopping – New Customer Registration

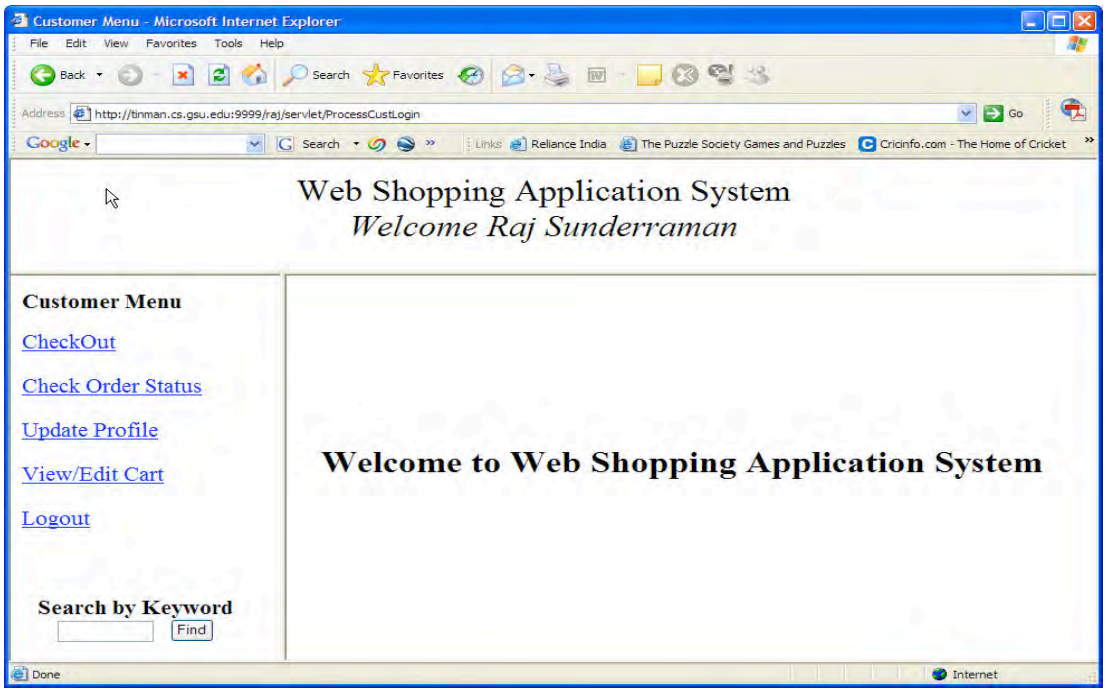

Figure 8.3: Web Shopping – Successful Sign-In 3-Frame Page

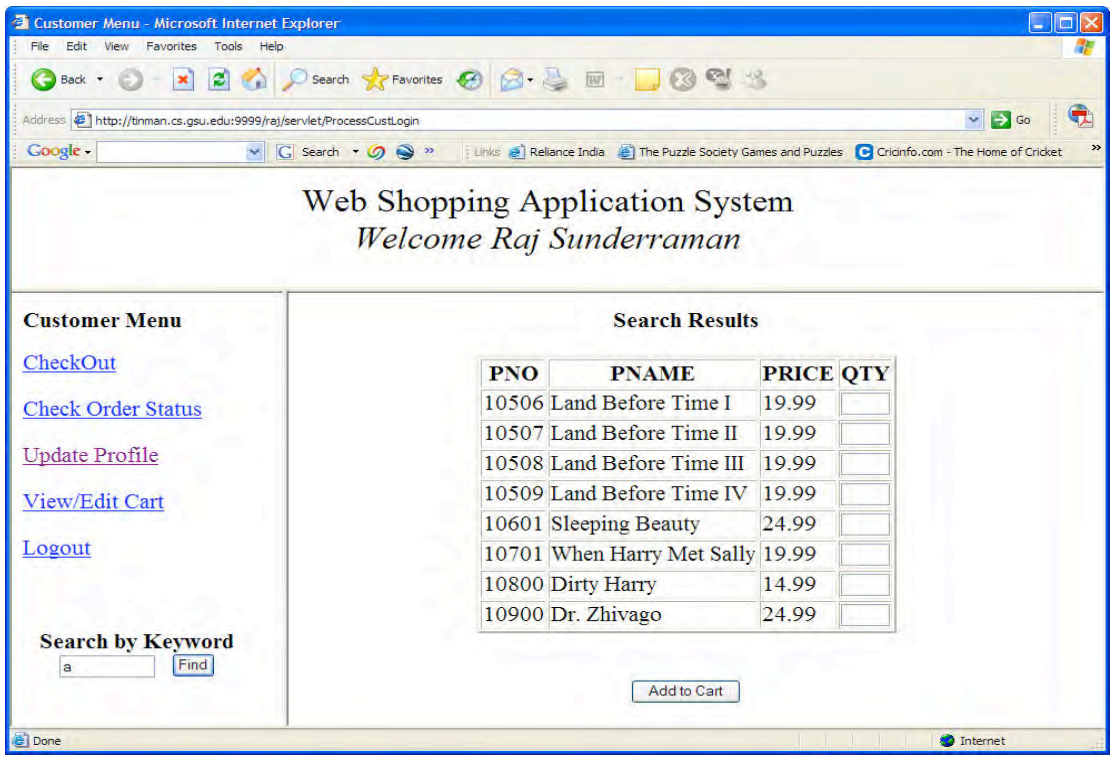

Figure 8.4: Web Shopping – Search Result Page

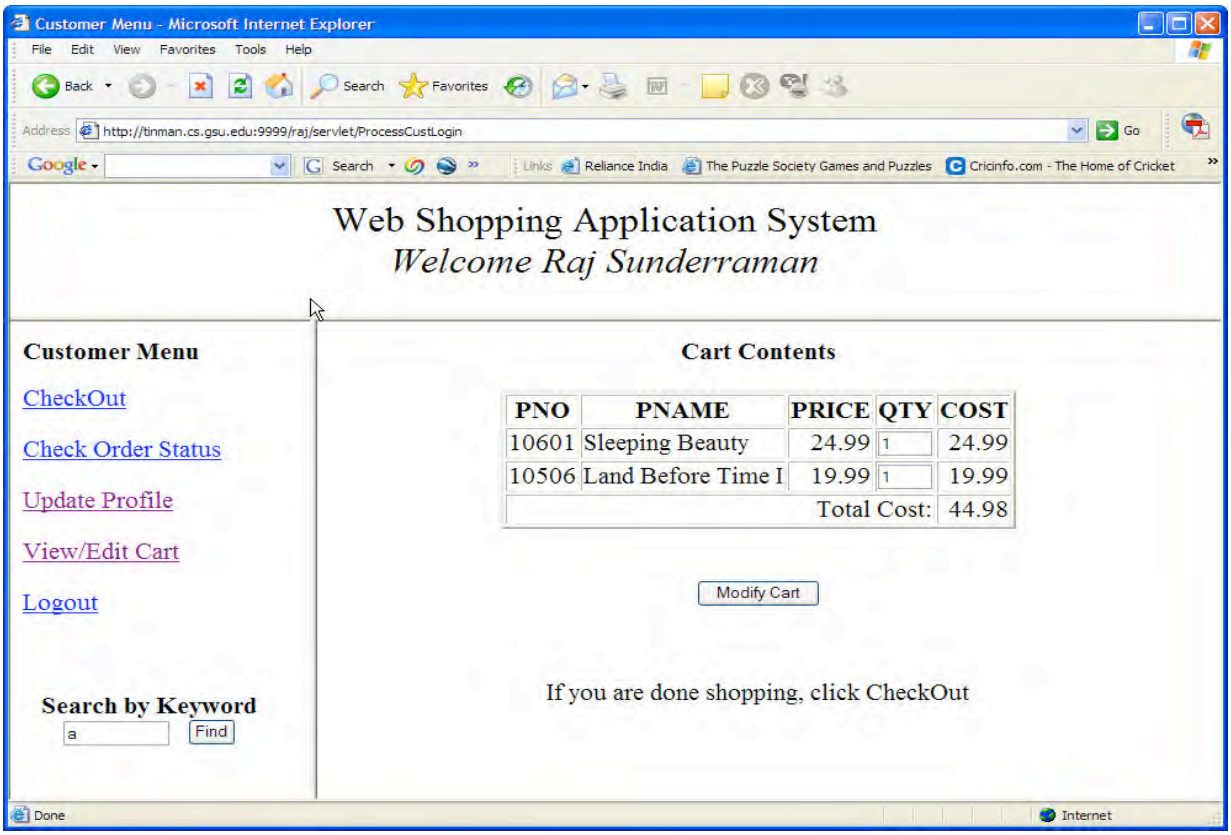

Figure 8.5: Web Shopping – View/Edit Cart

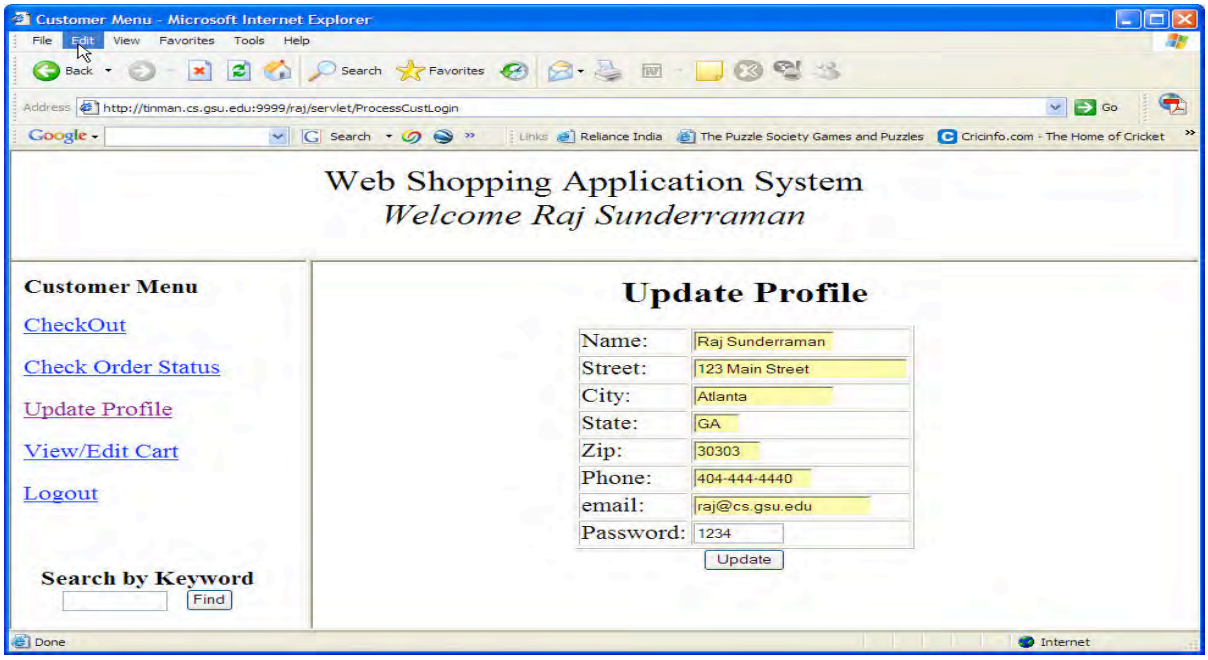

Figure 8.6: Web Shopping – Update Profile

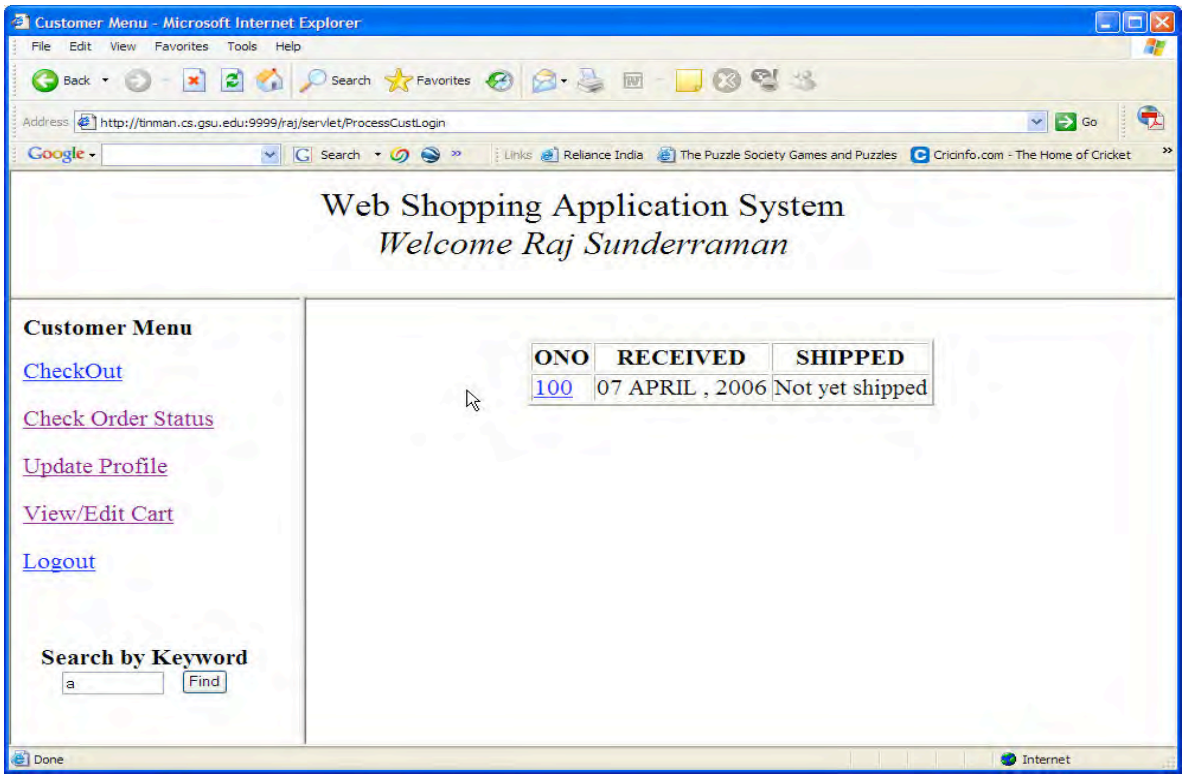

Figure 8.7: Web Shopping – Check Order Status

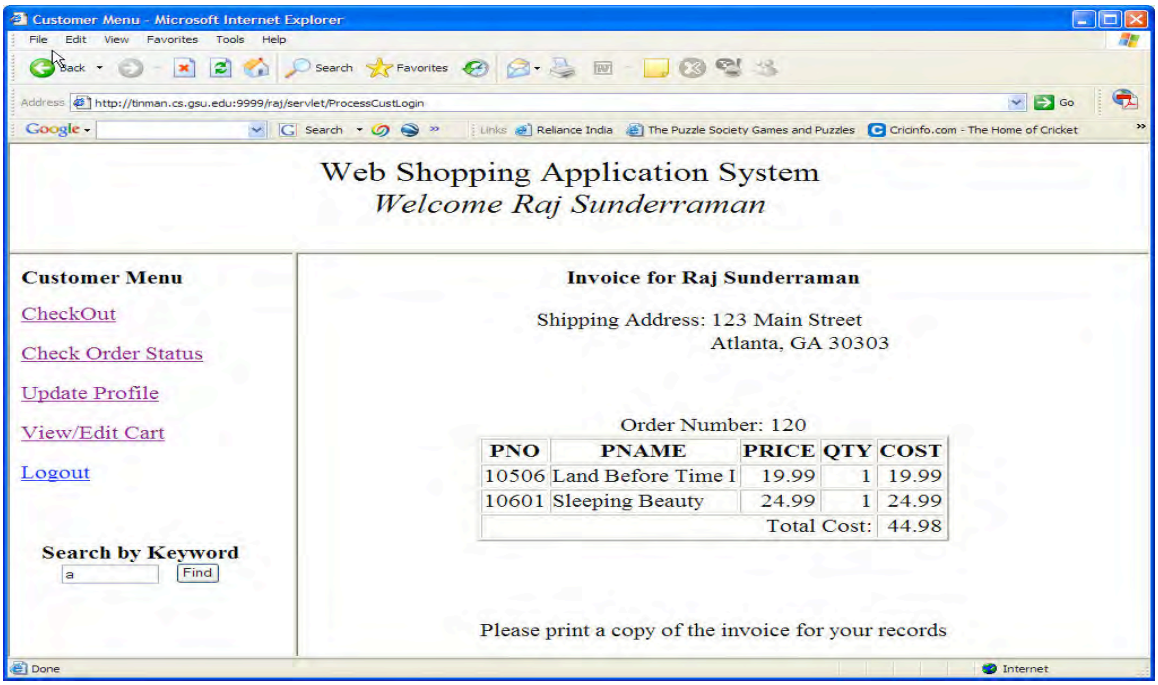

Figure 8.8: Web Shopping – Check Out

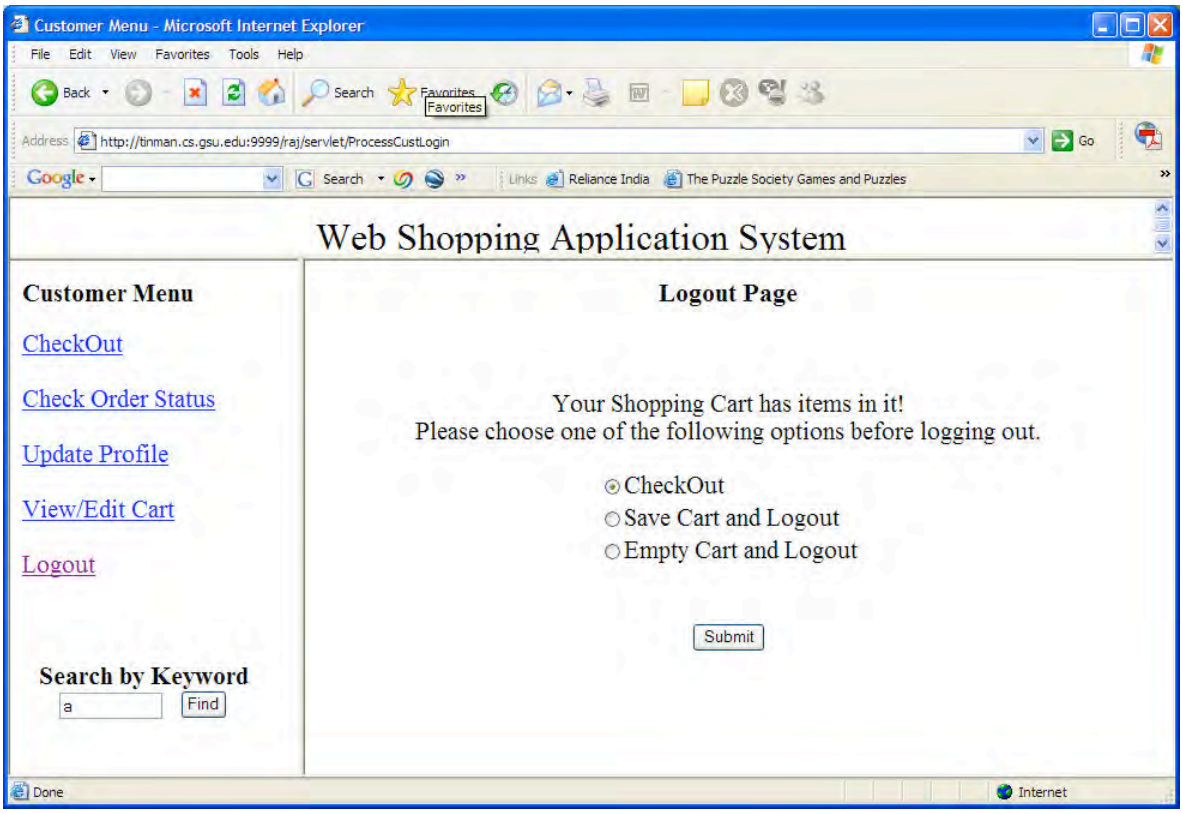

Figure 8.9: Web Shopping – Log Out

8.4 Online Bulletin Board System

Using PhP and MySQL implement an online bulletin board system that allows a set of authorized users to participate in an online discussion forum. The data for the bulletin board system should be stored in a MySQL database with the following schema:

```
create table bbusers (
  email    varchar(50),
  name     varchar(30),
  password varchar(10),
  nickname varchar(30),
  primary key (email)
);

create table postings (
  postId       integer(5) auto_increment,
  postDate     datetime,
  postedBy     varchar(50),
  postSubject  varchar(100),
  content      varchar(512),
  ancestorPath varchar(100),
  primary key (postId),
```

```
  foreign key (postedBy) references bbusers
);
```

The database has two tables:

1. `bbusers`: This table records information about users of the bulletin board. The `email` and `password` fields are used for signing into the system.

2. `postings`: This table records information about all postings as well as follow-up postings of the bulletin board. Each posting is assigned a unique `postId`. To keep track of the "tree-structure" generated by follow-up postings, the system keeps track of the path from root message to the posting in the `ancestorPath` attribute. The path is recorded as a colon separated list of posting Ids; for example the ancestor path `1:5:6:12` would indicate that the current posting has a parent posting with `postId=12`, a grand-parent posting with `postId=6`, a great-grand-parent with `postId=5`, and a great-great-grandparent with `postId=1`. With this structure, the entire bulletin board messages can be viewed as a collection (forest) of trees.

The Web application should implement the following basic functions:

1. User sign-in and sign-out.
2. Default display of messages in reverse chronological order and properly indented follow-up messages.
3. Post message and post follow-up message by user.

Figure 8.10 and 8.11 show possible user interfaces for the main display page and the follow-up display page.

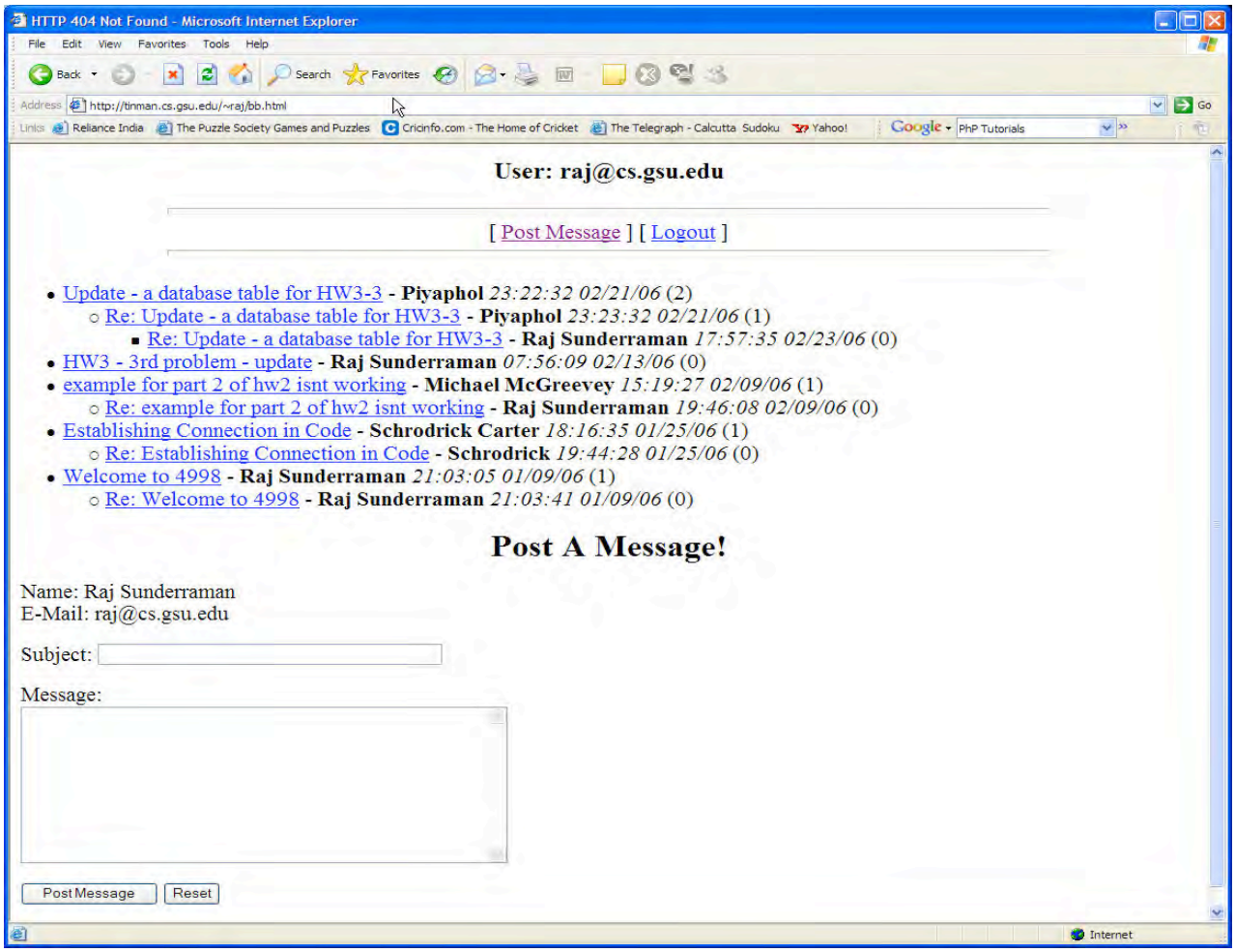

Figure 8.10: Bulletin Board – Main Display Page

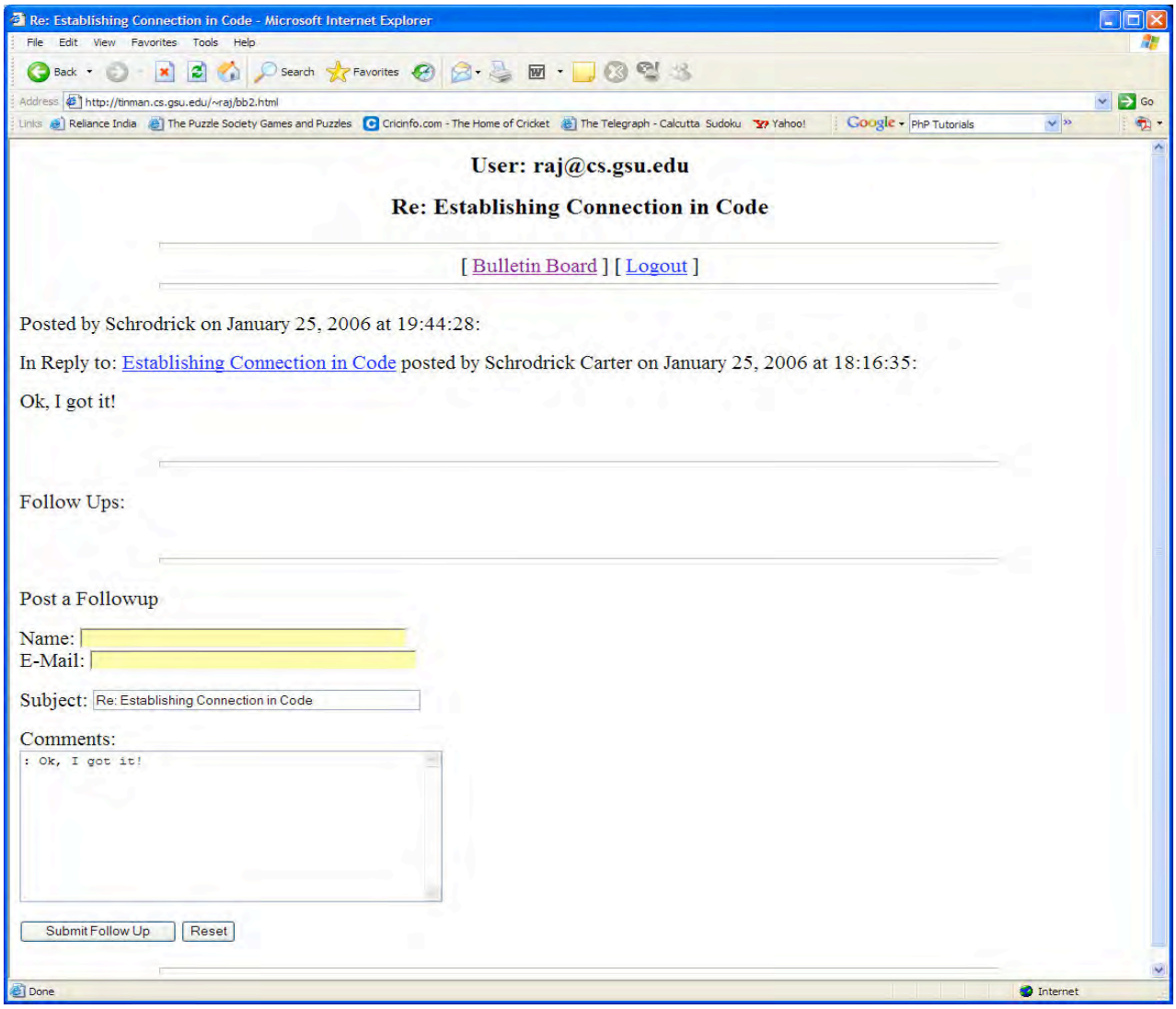

Figure 8.11: Bulletin Board – Follow-up Listing Page

8.5 Online Exam Management System

Using PhP and MySQL implement an online exam management system that allows (a) an administrator to create/delete/edit online multiple-choice exams and (b) student users to take these exams and view the results. The relational schema for the system is already designed and is shown below:

```
drop table exam cascade constraints;
create table exam (
  eno number(5),
  etitle varchar2(50),
  timeAllowed  number(8), -- minutes
  numberOfQuestionsPerPage number(3),
  primary key (eno)
```

```
);

drop table question cascade constraints;
create table question (
  eno number(5),
  qno number(5),
  qtext varchar2(2048), -- maybe be CLOB object
  correctAnswer char(1),  -- must be one of the options
  foreign key (eno) references exam,
  primary key (eno,qno)
);

drop table answerOption cascade constraints;
create table answerOption (
  eno number(5),
  qno number(5),
  ono char(1) check (ono in ('A','B','C','D','E')),
  optionText varchar2(256),
  foreign key (eno,qno) references question,
  primary key (eno,qno,ono)
);

drop table users cascade constraints;
create table users (
  uno number(5), -- primary key; system generated starting at 1
                 -- first user gets 1 and subsequent users get max+1
  email varchar2(64),  -- unique key used for signing in
  password varchar2(64),
  fname varchar2(64) not null,
  lname varchar2(64) not null,
  address1 varchar2(64),
  address2 varchar2(64),
  city varchar2(64),
  state varchar2(64),
  zip number(5),
  primary key (uno)
);

drop table enrolls cascade constraints;
create table enrolls (
  uno number(5),
  eno number(5),
  startTime date,
  finishTime date,
  foreign key (uno) references users,
  foreign key (eno) references exam,
  primary key (uno,eno)
);

drop table userResponse cascade constraints;
create table userResponse (
  uno number(5),
  eno number(5),
  qno number(5),
  response char(1)
    check (response in ('A','B','C','D','E','N')), -- N for No Answer
  foreign key (uno,eno) references enrolls,
```

```
  foreign key (eno,qno) references question,
  primary key (uno,eno,qno)
);
```

Here is some sample data for the `exam`, `question`, and `answerOption` tables:

```
insert into exam values (3,'Elementary History',10,3);

insert into question values
  (3,1,'The Battle of Gettysburg was fought during which war?','C');
insert into answerOption values (3,1,'A','World War II');
insert into answerOption values (3,1,'B','The Revolutionary War');
insert into answerOption values (3,1,'C','The Civil War');
insert into answerOption values (3,1,'D','World War I');

insert into question values
  (3,2,'Neil Armstrong and Buzz Aldrin walked how many \n' ||
       'minutes on the moon in 1696?','B') ;
insert into answerOption values (3,2,'A','123');
insert into answerOption values (3,2,'B','None');
insert into answerOption values (3,2,'C','10');
insert into answerOption values (3,2,'D','51');

insert into question values
  (3,3,'Which Presidents held office during World War II?','D');
insert into answerOption values (3,3,'A','Franklin D. Roosevelt');
insert into answerOption values (3,3,'B','Dwight D. Eisenhower');
insert into answerOption values (3,3,'C','Harry Truman');
insert into answerOption values (3,3,'D','Both A and C');

insert into question values
  (3,4,'In a communist economic system, people:','B');
insert into answerOption values (3,4,'A','Are forced to work as slaves');
insert into answerOption values (3,4,'B','Work for the common good');
insert into answerOption values (3,4,'C','Work from home computers');
insert into answerOption values (3,4,'D','Don''t work');


insert into question values
  (3,5,'Which president did not die while in office?','D');
insert into answerOption values (3,5,'A','John F. Kennedy');
insert into answerOption values (3,5,'B','Franklin D. Roosevelt');
insert into answerOption values (3,5,'C','Abraham Lincoln');
insert into answerOption values (3,5,'D','Ronald Reagan');
insert into answerOption values (3,5,'E','James A. Garfield');

insert into question values
  (3,6,'Which state refused to attend the Constitutional Convention \n' ||
       'in 1787 because it didn''t want the United States government \n' ||
       'to interfere with already established state affairs?','A');
insert into answerOption values (3,6,'A','Rhode Island');
insert into answerOption values (3,6,'B','New Hampshire');
insert into answerOption values (3,6,'C','New Jersey');
insert into answerOption values (3,6,'D','New York');

insert into question values
```

```
  (3,7,'Who founded Buddhism?','A');
insert into answerOption values (3,7,'A','Siddharta Gautama');
insert into answerOption values (3,7,'B','Jesus Christ');
insert into answerOption values (3,7,'C','Mahatma Gandhi');
insert into answerOption values (3,7,'D','Muhammad');

insert into question values
  (3,8,'Where is India?','D');
insert into answerOption values (3,8,'A','Australia');
insert into answerOption values (3,8,'B','America');
insert into answerOption values (3,8,'C','Africa');
insert into answerOption values (3,8,'D','Asia');


insert into question values
  (3,9,'What is the dominant religion in India?','B');
insert into answerOption values (3,9,'A','Islam');
insert into answerOption values (3,9,'B','Hinduism');
insert into answerOption values (3,9,'C','Christianity');
insert into answerOption values (3,9,'D','Buddhism');

insert into question values
  (3,10,'Near which river did archaeologists find India''s \n' ||
        'first civilization?','B');
insert into answerOption values (3,10,'A','The Tiber River');
insert into answerOption values (3,10,'B','The Indus River');
insert into answerOption values (3,10,'C','The Yellow River');
insert into answerOption values (3,10,'D','The Nile River');
```

The project should be implemented in two separate modules:

1. **Admin Module:** The admin module should allow administrators to create multiple-choice exams. This is basically a data input/update module that allows admin user to
    - **Create Exam**: After collecting top level exam details such as exam title etc, the user should be allowed to add questions one at a time. The add question screen should contain input boxes and text areas for top level question data and a pull down list for number of options (2 through 5). Using Javascript, you should create the right number of answer option data input text areas for answer option text along with a check box that can be checked for "correct answer" option. Once all information is given, the user can submit the question to be added to the database. The user should be presented with the add question screen in case they want to add the next question.
    - **Delete Exam**: Given a list of exams, the user chooses exam to be deleted. Only exams in which no one has signed up should be presented. A "confirm delete" screen should be presented before the exam is deleted.
    - **Edit Exam**: The user should be able to add new questions at a particular position and delete a question.
2. **User Module:** The user module should allow ordinary users to register, sign in, update profile, sign up for exams, take exams, and see their results.

- **Register, Change Password, Sign In, and Sign Out**: A standard login page (with email and password text boxes) along with a "If you do not have an account, register here" link. Once logged in successfully, the user should be presented with several options including "Update Profile" in which they can change some of the data about themselves such as password, address etc.
- **Enroll in Exam(s)**: A menu option for the user - used to enroll in a particular exam (you may present a select list of all available exams and ask the user to choose one). Note: If the student is already enrolled in the exam and has finished taking the exam or is currently taking the exam (i.e. has started taking the exam but not yet finished), a warning should be issued stating that his answers will be reset. You may confirm that the user wishes to reset the old exam. Once enrolled, you should present the user with a confirmation which includes details about the exam he or she has just signed up for.
- **Take an Exam**: The user should be presented questions from where they left off the last time they signed on to take the exam. Questions should be presented in order using the pre-defined number of questions per page. Once answers are submitted, they cannot be revisited. The user is then presented the next set of questions until time runs out or there are no more questions.
- **View their Grade Report(s)**: The user chooses the exam for which they like to view results. Only list of exams that have been completed should be presented. The format of the grade report is up to you, but must include number of questions answered correctly, total number of questions, percentage coorect, and a detailed listing of user responses and correct answers.

8.6 Online Auctions

Using PhP and MySQL implement an online auction website (AuctionBase). The relational schema for the system is already designed and is shown below:

```
drop table member cascade constraints;
create table member (
  mid varchar2(10) not null,
  email varchar2(40) not null,
  fname varchar2(20) not null,
  lname varchar2(20) not null,
  street varchar2(50) not null,
  city varchar2(30) not null,
  state varchar2(20) not null,
  zip number(5) not null,
  phone varchar2(12),
  password varchar2(20),
  primary key (mid)
);
--
drop table category cascade constraints;
create table category (
  cname varchar2(120),
  primary key (cname)
);
--
```

```
drop table item cascade constraints;
create table item (
  ino number(5),
  title varchar2(128) not null,
  category varchar2(120) not null,
  description varchar2(2000),
  openDateTime date,
  sellerId varchar2(10) not null,
  startingBid number(7,2) not null,
  bidIncrement number(7,2) not null,
  closeDateTime Date,
  winnerId varchar2(10),
  primary key (ino),
  foreign key (category) references category,
  foreign key (sellerId) references member,
  foreign key (winnerId) references member
);
--
drop table bid cascade constraints;
create table bid (
  ino number(5),
  buyerId varchar2(10),
  bidPrice number(7,2),
  timeOfBid date,
  primary key (ino,buyerId,timeOfBid),
  foreign key (ino) references item,
  foreign key (buyerId) references member
);
--
drop table rating cascade constraints;
create table rating (
  ino number(5),
  buyerRating number(1) check (buyerRating between 1 and 5),
  buyerComment varchar2(100),
  sellerRating number(1) check (sellerRating between 1 and 5),
  sellerComment varchar2(100),
  primary key (ino),
  foreign key (ino) references item
);
```

Initial data is given below:

```
insert into member values
  ('a100','a@cs.gsu.edu','Tom','Jones','120 Main Street','Atlanta','GA',30303,
   '404-111-1110','a123');
insert into member values
  ('m100','m@cs.gsu.edu','Jim','Smith','121 Main Street','Atlanta','GA',30303,
   '404-111-1111','m123');
insert into member values
  ('p100','p@cs.gsu.edu','Don','Fleming','122 Main
Street','Atlanta','GA',30303,
   '404-111-1112','p123');
insert into member values
  ('q100','q@cs.gsu.edu','James','John','123 Main Street','Atlanta','GA',30303,
   '404-111-1113','q123');
insert into member values
```

```
  ('s100','s@cs.gsu.edu','Monty','Jones','124 Main
Street','Atlanta','GA',30303,
   '404-111-1114','s123');
--
insert into category values ('Books:Biology');
insert into category values ('Books:Computers');
insert into category values ('Books:Economics');
insert into category values ('Books:Fiction');
insert into category values ('Computers:Apple:Desktops');
insert into category values ('Computers:Apple:Laptops');
insert into category values ('Computers:PCs:Desktops');
insert into category values ('Computers:PCs:Laptops');
insert into category values ('Computers:Storage:Hard Drives');
insert into category values ('Computers:Storage:Flash Drives');
insert into category values ('DVDs:Action');
insert into category values ('DVDs:Comedy');
insert into category values ('Music:Blues');
insert into category values ('Music:Jazz');
insert into category values ('Music:World');
insert into category values ('Video Games:Systems:XBox 360');
insert into category values ('Video Games:Systems:Wii');
insert into category values ('Video Games:Systems:Playstation');
insert into category values ('Video Games:Systems:Nintendo DS');
insert into category values ('Video Games:Games:XBox 360');
insert into category values ('Video Games:Games:Wii');
insert into category values ('Video Games:Games:Playstation');
insert into category values ('Video Games:Games:Nintendo DS');
--
insert into item values
  (1000,'Mario Party IV','Video Games:Games:Wii','Excellent Condition ' ||
      'Best Seller; Super Graphics',
      to_date('21-APR-2008 1700','DD-MON-YYYY HH24MI'),
      'a100',20.00,2.00,
      to_date('28-APR-2008 1700','DD-MON-YYYY HH24MI'),
      null);
```

The project should be implemented in the following stages:

Stage I: Implement the "Browse/Search" part of the AuctionBase website.

Each Web page in this part should have a "Top" portion which contains:

- A "Bread Crumb" indicating the level of the category being browsed. For example, the initial page should have HOME as the bread crumb. Lower levels of categories would have bread crumbs such as HOME::DVDS::FICTION, HOME::DVDS, HOME::BOOKS, HOME::BOOKS::ECONOMICS, etc. Each of these terms in the bread crumbs should be hyper-linked so that when they are clicked, the page refreshes and shows the level that is clicked.
- A "Search" text box and a pull-down list of top-level categories and a submit button. This should allow users to search for items using keyword. The results of the search should be shown in list form.

The Web page should contain a "Bottom" portion which lists either the sub-categories for the rightmost category in the bread crumb or a list of items if the rightmost category in the bread crumb is the lowest level category. These sub-categories and items should be hyper-linked as well. The sub-categories should be hyper-linked to the next level page and the items should be hyper-linked to a "Detail" page for the item.

The "Detail Page" for the item should list all details of the item and should provide a text box for the user to enter a bid and a submit button.

Stage II: Implement the following functions:

1. Login/logout.
2. Update member profile.
3. Place a bid.
4. View closed items along with open items. This should be be displayed along with browse/search options, but with no text box/submit button for "bid".
5. Place feedback.
6. View feedback/ratings for a particular member.