

<p style="text-align: center;">SYSTEMS DESIGN / CAPSTONE PROJECT MIS 413</p>

Remaining Tasks/Individual Responsibility to Complete the Project!

For the month of April, we want to complete our project and have a completed prototype by team.

Each Team should review the remaining tasks for the project and each team member should volunteer for one of the remaining tasks. Each team should send me one email indicating who is doing each task (i.e. Task A: Linda; Task B: Jeremy, etc); include all team members in the email cc. Send this email by the end Wednesday, April 12.

See the end of this PDF on what work should be accomplished for project 8 and the work that should be accomplished for project 9.

Here are some hints on the work that needs remaining:

Project: Capstone Tracking

- A Build the screens to support the capstone project table and its CRUD, items would include capstone project, title, type, status, proposal date and location
- B Build the screens to support the committee members on a capstone project, included would be the projectID, the committee members and indicating the chair.
- C Build the screens to support the 'validCourses' and the studentRequiredCourses, included would be adding / updating / deleting which courses (and major) qualify and then the 2nd table for the courses that a particular student has taken and when.

Project: 4+1

- A A Build the screens to support the Application Table and its CRU. Do all items except for the appointment date / completed items
- B Build the screens to support these two items: validCourses (which courses qualify), and reports – build a listing of all students and the information from the application table.
- C Build the CRUD to support studentCourses (which courses has the student completed and when)
- D Build the processes to permit the admin to accept students into the MS CSIS Program and also archive them.

Project 8 Deliverables:

Everyone - Provide Evidence that your Cookie was built in project 7

1. On you Menu.aspx page – add a label before the </asp:Content>, tag this label will display cookie info, name this label _cookieRetrieveInfo, change the text property to "".
2. On your **Menu C# page** -- Build the following sub/void/methods immediately **before the last two right brackets - }**. Please remember you can not copy and paste directly from this page into your C# page.

```
protected int RetrievePersonID()
{
    // this checks to see if a cookie exists and returns the PersonID
    int iTempID = 0;
    if (Request.Cookies["userInfo"] != null) //note the != represents Not
    {
        iTempID =
        Convert.ToInt32(Server.HtmlEncode(Request.Cookies["userInfo"]["personID"]));
    }
    return iTempID;
}

protected int RetrieveRoleID()
{
    // this checks to see if a cookie exists and returns the RoleID
    int iTempID = 0;
    if (Request.Cookies["userInfo"] != null) //note the != represents Not
    {
        iTempID =
        Convert.ToInt32(Server.HtmlEncode(Request.Cookies["userInfo"]["roleID"]));
    }
    return iTempID;
}
```

3. We now need to tell the menu.aspx page to execute these methods/functions as it builds the page, thus: add in the **page_load** subroutine/void a call to the new methods and populate your label

```
this._cookieRetrieveInfo.Text = "Your Person ID is: " +
    Convert.ToString(RetrievePersonID()) + " and your Role ID is: " +
    Convert.ToString(RetrieveRoleID());
```

4. Place the Admin items on your menu in a panel and make the panel's visible property to be false. (basically we will only show the admin items to people's roles who are admins)
5. Also on the MENU C# page in the **Page Load Method**, after you have retrieved the cookie values, build a series of IF and else IF's to turn on the appropriate panel(s) based on their role,
 - a. i.e if (RetrieveRoleID() == 1)
 - i. then make the admin menu items visible

Project 8 and 9 deliverables by project and responsibility segment

Capstone:

- A. Project 8:
 - a. Build a menu item to go to a new page in the user folder, name the new page "CapstoneTracking"
 - b. The CapstoneTracking page will mimic the Profile page, in terms of a panel, insert items and a grid
 - c. For project 8, build the grid and make it support the select, update and delete items,
 - i. The grid should error check
 - ii. The grid should look professional
 - iii. The grid should provide valid update or deletion messages
- A. Project 9:
 - a. Build the insert portion of the CapstoneTracking page
 - i. Likewise, error checking, messages, toggling of the panel is required
 - b. Build a new menu item under the admin menu, and build a new page in an admin folder (new). Name the new page Tracking
 - c. On this new page, display in a grid all students who have not completed their project in last name order.
- B. Project 8:
 - a. Build a new menu item under the student menu. Place this new page in the user folder and name it "Committee". The page should mimic the profile page as it will have an insert panel and grid.
 - b. For Project 8, build the grid and make it support the following options:
 - i. Delete, permit the deletion of a committee member
 - ii. Update – ONLY the Chair field should be updatable, if you want to change a committee member you would delete the old and then insert the new. The chair column should have a drop down box stating yes or no in edit mode.
- B. Project 9:
 - a. Add the functionality to insert a new committee member:
 - b. The 'insert' panel items should have two drop downs
 - i. One for a list of faculty (Not already on the committee)
 - ii. One for a Yes/No drop down on if they are the chair
 - c. Toggle panels/grids as on your profile page
 - d. Add a new admin menu –name it CommitteeOverview and place the new page in an admin folder (new)
 - e. This new page should be nothing but a select statement to show all students, project status, and committee members. Only display projects that are not final. Sort by Student Last Name. Select Only.

C. Project 8:

- a. Build a new menu item under the admin menu. Place this new page in the admin folder(new folder) and name it validCourses. The page should mimic the profile page as it will have an insert panel and grid.
- b. For project 8, build the grid and make it support only the select, delete items,
 - i. The grid should look professional
 - ii. The grid should provide deletion messages
- c. Build the insert panel items that error check and provide the valid completion messages

C. Project 9:

- a. Build another menu item in the student area, it should take you to another new page in the user folder, name this new page requiredCourses. It should also mimic the profile page
 - i. The grid should permit only select and delete option and valid message
- b. Build the insert panel on the above page, with a series of drop downs to permit a student to indicate the courses and dates for their courses.
- c. Build another new menu item under Admin, pointing to a new file named: coursesCompleted, then build a grid that displays (only select) the courses they have completed and when. Only display those students not complete.

4+1 Project:

A. Project 8:

- a. Build a menu item to go to a new page in the user folder, name the new page - Application
- b. The Application page will mimic the Profile page, in terms of a panel, insert items and a grid
- c. For project 8, build the grid and make it support the select, update and delete items (do not do the appt date/completed columns),
 - i. The grid should error check
 - ii. The grid should look professional
 - iii. The grid should provide valid update or deletion messages

A. Project 9:

- a. Build the insert portion of the Application page
 - i. Likewise, error checking, messages, toggling of the panel is required
- b. Build a new menu item under the admin menu, and build a new page in an admin folder (new). Name this new page: Applicants
- c. On this new page, display in a grid all students who have applied to the program. Display key information. No updating or deleting

B. Project 8:

- a. Build a new menu item under the admin menu. Place this new page in the admin folder (new) and name it "validCourses". The page should mimic the profile page as it will have an insert panel and grid.
- b. For Project 8, build the grid and make it support the following options:
 - i. Delete, permit the deletion of a valid course
- c. Complete the functionality of the validCourses to be able to insert a new course.

B. Project 9:

- a. Build a new menu item under the admin menu, and build a new page in the admin folder, name it 'appointments'
 - i. This page should display two objects, one grid for students with appointments
 - ii. The second object should be a drop down list of students without appointments
 - iii. Once you select a student from the drop down, you should display two text boxes to permit the admin to enter a date and time for an appointment, then update the grid and dropdowns.

C. Project 8:

- a. Build a new menu item under the student menu. Place this new page in the user folder and name it courseComplete. The page should mimic the profile page as it will have an insert panel and grid.
- b. For project 8, build the grid and make it support only the select, delete items,
 - i. The grid should look professional
 - ii. The grid should provide deletion messages
- c. Build the insert panel items that error check and provide the valid completion messages. The insert should be a drop down list of valid courses (not in the grid for that student)

C. Project 9:

- a. Build another new menu item under the Admin panel, pointing to a new file in an admin folder (new) named: coursesCompleted, then build a grid that displays and display (only select) the courses they have completed and when.
- b. Build another new menu item under the admin panel and point to a new page in the admin folder, name this page 'statusChange'.
 - i. This new page should have the names of all students in the application file who have not be accepted, once you click on their name in the grid, it should give the admin two options, one to *accept* the individual or one to change the status *not accepted*.

D. Project 8:

- a. Add two additional fields to the application table: Accepted (bit) and dateAccepted (date/time). Set a default value for the accepted field to 0 (or false), and populate any current applications in the table as false (not accepted)
- b. Build a new folder in your project, name it admin
- c. Build a new page in this admin folder, name it AcceptStudents
- d. Build a link on the menu page (under admin options) to the page you just built
- e. On this new page, use a Checkbox List to display those students who have NOT been accepted.
- f. See User Guide 5.2 to learn how to build a checkbox list. You will need to create a stored procedure that pulls the names and personID of all applicants not accepted.

D. Project 9:

- a. Add another new page in the admin folder, name it Archive
- b. Build a link on the menu page to take you to this new page
- c. The Archive page, should have a button that says, "Archive Accepted Students"
- d. Create a new table in the database, name this table: studentArchive
 - i. Columns should be personID, FirstName, LastName, Email and dateAccepted
- e. Once the button is clicked, then you should move students in the application table to the archive table for those where Accepted = true
- f. Build a grid on the bottom of the page, that shows all students archived in last name order