FINAL PROJECT
[250 POINTS TOTAL]

Out: 10/15/2014
Session 7

Due: 11/02/2014
@ 6:00 pm

INTRODUCTION

The final project is an important part of this course. It accounts for 25% of the grade and is intended to encompass the full course content. **The objective of the final project is to design, build, populate, and create a server-based relational database application and to develop a Web-based client application to interface with that database**.

DUE DATES

Project Proposal due on **November 02, 2014 at 6:00 pm.**

Final Project Report due on **December 17, 2014 at 5:30 pm.**

GUIDELINES

Choose your own database problem. A particular problem that is relevant to you will be far more interesting for you to develop. If after giving some thought to it you cannot devise a database problem, consult with your instructor or the teaching staff for assistance. Do so as early as possible.

Build your own Client/Server application. Developing your own ideas about a client application should present an interesting challenge. Limit your application in order to keep the project manageable and achievable within the time we have allotted this term. While you need not plan on building all the functionality that we have developed in class using ColdFusion, you should devise some meaningful functionality. Relevant operations (data retrieval and modification) should be done via the client interface.

Work on an original project. The project you submit for this class must be your original work. You may not submit a project that you have already submitted for another class, regardless of whether that class was taken at Harvard Extension School or any other institution. **Your project report must include the following statement:** ***"I hereby certify that this project was prepared especially for this course, and that this or a similar version has not been submitted to any other course."*** If you have received permission from the instructor to do otherwise, you must explain it here.

Use the Class Server. Even if you are building the project for someone who does not intend to use **Oracle** and/or **ColdFusion**, you must develop the application using those technologies, and then translate it into different software to fit the needs of your client after the course ends. In addition, your project must run on the class server. **Projects developed and submitted in another platform and/or using alternative technologies will receive no credit.**

Devise a database problem that is about the magnitude of the Wiggly, Inc. database, i.e. about four to six tables with some quantitative data, that is, data other than just text. Numeric data allows for interesting queries. You may enhance your design with reference tables. **The reference tables do not count towards suggested limit of four to six tables**.

Remember that the first task is to design the database. Think about the kinds of **queries, forms and reports that might be useful with your application** and let that consideration determine the data in the database. **You are expected to include appropriate examples of competent information management that we have covered in class,** for example, appropriate field data types and lengths, check constraints, validation rules, analytic queries, multi-table forms, reports, and a main menu to your application.

Develop your own user interface. While following the same style used by the professor (CSS) in the assignments is a good idea, you must develop your own look and feel for your final project. Refrain from using proprietary information or designs developed by third parties. If you believe there is a reason to do so, please discuss it with the teaching staff and obtain permission **before** attempting to implement someone else's interface.

Students may team up in-groups of no more than two students to propose and submit a final project. I expect, however, that a team project will be a more rigorous project, including more tables or more data or a more robust client application. In case of team projects, the final project submission must include a detail of the contribution made by each member of the team.

Students are encouraged to attend TA sessions in order to discuss their final project with the teaching staff. This is important at every stage of the project. For example, you may want to discuss your initial database design before submitting the proposal, or you may want to discuss your database after you have built the database and accomplished some basic functionality on your client application. These sessions will help make sure that you are making progress on the project.

The late policy does not apply to the final project. The final project report must be handed-in on or before **December 16, 2013**. Final grades will be calculated on the basis of completed assignments, the midterm, and the final project. Zero credit will be given for unfinished work.

Extensions will be granted only for the most extenuating circumstances and generally only to complete the final project. You must petition the class instructor for an extension on or before **November 30, 2014**.

DELIVERABLES

Project Proposal

The project proposal should be **about 2 pages long plus the database prototype**. Please, be generous with spacing. Remember, presentation matters! There should be no grammatical errors. You may download a sample proposal available in the Problem Set section of the course Web site.

Your proposal **must** include the following aspects:

Statement of the Problem and purpose of the Project. Why is this application important? There are a number of reasons why any particular application might be important. For example, your company might be keeping track of a process by using a Microsoft Excel spreadsheet. Describe the current situation to the best of your abilities.

Scope of the Study. What are the specifics of the project? Briefly describe the forms and reports you plan to include on your application. Quite often, students tend to underestimate the time it will take for them to complete the project. Remember you will only have a couple of weeks to complete this project; therefore you should set the boundaries of your work. Even if you modify the selection of forms and reports, it is a good idea to set the boundaries early in the project.

Methodology. How will you collect the information to complete your project? Does your database require interviewing your company staff, or do you have complete access to the information required to build the database?

Prototype. You must include the **ERD** and **record diagrams**. You may do so by using any drawing tool or even a Microsoft Access database diagram. The prototype should include fully normalized tables with defined Primary Keys.

You should receive an approval email within 10 days of receipt. This will give you plenty of time to work on your project. The email will include any suggestions which will assure that you have a solid database design upon which to proceed with creating you application. Attempting to complete a final project without a good database design usually results in frustration and an unsatisfactory final project.

Final Project

For the final project you should submit the following:

Access information to your application, including location, and logins and passwords required to access your application. If there are any special instructions on how to run or access your application, you must include them as well. This information must be at the top of your report, and follow the example below:

URL:	http://cscie60.dce.harvard.edu/~jharvard/fp/index.cfm
User:	user / 60user
Administrator:	admin / 60admin

The example above indicates that John Harvard has created two test accounts: one with user privileges and another one with administrator privileges. For ease of grading, it may be easier to use the same user and administrator information indicated above. **Projects submitted without this information at the top of their reports will receive no credit, even if they are the ones indicated above.**

A one to two page description of the application you have built including a brief description of the purpose of the application and description of any forms, queries, reports and menus you have built. You should be able to reuse some of the information provided in the proposal and adjust it accordingly.

A listing of bugs you were unable to fix, and an action plan on how to fix them. If you are forthcoming in describing the issues we will face, you are less likely to lose too many points. Bugs your graders find when testing your application are those likely to cost you more points.

Special features you implemented. A portion of your grade is based on the implementation of special features (see below). You will receive points for them as long as the features work and they are documented in the report. The TAs will not be scouting for them.

A full data model of your database design including record diagrams as we have done in class. This is, you must include the ERD and record diagrams. Do not submit any of the intermediate steps in ERA database design.

The database schema including the create table statements, and the select * from each table. You may use the DDL sample we presented in class as a template.

The application code (e.g., ColdFusion .cfm code).

Screen shots of some of the screens of your application, showing database interaction. Think of it as a walk through of your project.

Make sure that all the pieces work together before embarking on the finished result. For example, if your application requires an email or a file download capability, try out that functionality early in your application development. While we do not expect you to become an expert with the systems you choose, the idea is to demonstrate at least the course content through a sound example of a Web-based database application.

GRADING GUIDELINES

Grading the final project will emphasize the following:

Project Proposal (20 points)

The proposal should clearly state the problem your information system is expected to solve, describe it accurately, and present it in good taste.

Database design and Implementation (100 points)

The database design should represent fully normalized tables with effective table relationships and enforced referential integrity. Data integrity should be enforced in the form of validation rules, integrity constraints, and appropriate datatypes. There should be sufficient data in every table to demonstrate the full range of information management needs of this application. The implementation should include at least two of the following: triggers, transactions, procedures. For example, (a) two triggers, (b) one trigger and a procedure, or (c) two transactions. You must describe those in the final project report in order to receive credit for them.

Client Application (80 points)

The database application should offer presentable, user friendly, and informative features which allow the information to be managed while insuring that information inserted/updated/deleted does not violate any integrity constraints you have built into the database design. For example, consider implementing drop down lists (combo boxes) to your forms to offer a choice where the user would have little chance of knowing the set of acceptable values. The application should provide a presentable user interface, in the form of a menu. A good way of testing this portion is to ask a friend to test it without providing him any explanation of what to enter. Experience tells us many students do poorly in this section.

Special Features (25 points)

The application must include special features for enhancement. For example, adding a search tool to your application, or an email or file download functionality. **The definition of a special feature is one that was not required for any assignment.** The number of features will depend on the type of features you implement. Quality is more important than quantity. For example, securing your application is more time consuming than adding a search tool. Securing your application will most likely be enough, but implementing a search tool will need one or two additional special features. Use your best judgment. You may have to do some research to satisfy this requirement. **To earn credit, your report must include a description of the special features.**

Report (25 points)

The final report should clearly state the problem your information system solves, describe it accurately, list the most relevant queries, triggers, transactions, procedures, and special features, basics of the client application, and present it in good taste.

SUBMITTING YOUR PROJECT

Project Proposal

Zip your document(s) using your Unix login name, and the assignment number. For example, **John Harvard** would save his file with the name **jharvardfp1.zip**, **Woody Woodpecker** would save his file with the name **wwoodpecfp1.zip**, and **Sue Li** would save her file with the name **slifp1.zip**. **You may lose up to 10% of your grade if you don't use the indicated naming convention.**

Final Project

Zip your document(s) using your Unix login name, and the assignment number. For example, **John Harvard** would save his file with the name **jharvardfp2.zip**, **Woody Woodpecker** would save his file with the name **wwoodpecfp2.zip**, and **Sue Li** would save her file with the name **slifp2.zip**. **You may lose up to 10% of your grade if you don't use the indicated naming convention.**

Submission

Submit the zip file in the corresponding dropbox located on the Course Web site.