Final Project Due Date: Sunday, November 27th at midnight Server/Database Lockdown Time: **11:59 PM**
Course Web Page: https://blackboard.missouri.edu

CS 3380 FALL 2016 Final Group Project: AIRLINE RESERVATION SYSTEM
Groups: Should be composed of 4 – 5 students

Objectives:
- ✓ Create a website for establishing an airline reservation system. Your group should agree on a standard look-and-feel and theme for the site, to which all members will adhere, and deploy the site on one of your tc.rnet accounts.
- ✓ Design an entity relationship diagram using crows foot notation and a database schema for the backend of your website.
- ✓ Work together as a team to complete the project.
- ✓ Seamlessly integrate various group members' work as well as different aspects of the web application into a single cohesive web application.
- ✓ Utilize relative sequencing and automating web logging activities.
- ✓ Test individual components of the site by populating the database with test data.
- ✓ Produce a coherent report that details the full design, including ERD, user-interaction, multiple user login credentials (at least one for each type of user), algorithms used, and a basic user's manual.

Brief Introduction
Most of us have had to use airline reservation systems at one time or another. In this project you will design a system for an airline named Missouri Air or "MA". This is a new start up company and you will have to read the requirements of the system (including business rules) and develop a fully functional system complete with documentation. I will provide some "high level" requirements but it is YOUR job, as a team, to ask any questions of me to make sure your project meets MY specific requirements. Read these "business rules" and design your database and application accordingly:

- ✓ Employees – there are 3 types: Administrator, Pilot and Flight Attendant. You MUST use a hierarchical (super/sub) entity for this.
- ✓ Customers – only one type of customer. They make flight reservations
- ✓ Equipment – airplanes have different attributes including seating capacity, "N" numbers from the US Government, types (737, 747, MD80, 767, RJ4, RJ12, etc.), # of attendants required, # of pilots required, etc.
- ✓ Flights – these consist of a flight number (Flight 232, etc.), they must have an aircraft assigned, they must have pilots assigned, flight attendants assigned, a departure city (FROM) and a destination city (TO).

Specifications
The web application should support four different user types (administrator, pilot, attendant, customer) and allow access accordingly. The functionalities for these roles are defined below. Only three of these will require authentication (administrator, pilot, attendant).

"CUSTOMER" USERS
Customers should be able to browse/search the global directory of available flights or search/sort by

a) Departure city (FROM)
b) Destination city (TO)
c) Day/Time of available flights
d) Price ($0 to $999, etc.)

Selecting criteria from the above should send the customer to a list page that displays all the flights in the database that satisfy their selection criteria and the individual information for each flight including dates, times, from, to, price. When a customer selects a reservation they must enter first name, last name, age, number of bags. The price for the flight will be calculated as the price in the database PLUS $20 for each bag and 5% sales tax. This final information should be displayed to the customer and they have one final chance to approve or discard the reservation. Once they "approve" then the customer record is created and all appropriate tables are updated.

"PILOT" / "ATTENDANT" USERS

1. Pilots should, of course, register to use the web application and must provide their username and password to access the system. You should store all the basic information about a pilot including status (Active or Inactive), total hours of flying, rank (co-pilot or officer), equipment certified to fly (737 or 747 or MD80 or RJ4 or all of them?). There should be some means provided to allow a pilot to edit their information. Pilots should also be able to view a log of all their past and current flights they have made.
2. Flight attendants should have the same access as pilots but different information such as rank (senior or junior). They do NOT need total hours of flying or equipment certified to fly on.

"ADMINISTRATOR" USERS

Administrators will have full access to the site. They should be able to edit information on any user (pilot, attendant, customer), reset passwords, monitor all flights, add pilots, attendants, equipment and flights to the master list when needed.

AUTHENTICATION

You should write a generalized module to perform user authentication against a database table of "userIDs" and encrypted passwords. This module should handle form-layout, logins, logouts, and session management of "userIDs.". Pilots and attendants will need to be authenticated.

WEB LOGGING

The site should have extensive logging and log utilities for ALL users. All logging should be in a single log table. Logs should contain information like IP address, date/time of action, what SPECIFIC action was taken, who performed the action, etc. Administrators will be able to see the entire log, while pilots, attendants and customers will only see items in the log related to them. The page that allows a user to view the log should include means to filter the log so that specific information may be located quickly (for example, by date of action, by type of action [reservation, flight, etc.], number of log entries to display, etc.).

Notes and Other Requirements

1. Authentication tables MUST store the passwords in ENCRYPTED form. You will lose 15% of the total points for storing passwords as clear text in the database. I suggest MD5 as a minimal level of password hashing. The authentication table should be separate from the user table and should contain ONLY userIDs, passwords, and roles.
2. You should user triggers/procedures/functions wherever possible (and you MUST use them where indicated). Relative sequencing with will be discussed further in class. It is advised that you implement the logging capabilities.
3. All your webpages should use a common color scheme. Do NOT use plain black text on a white background.
4. All programming or script code should be modular, to increase the readability and to maximize the reuse of code fragments. You should write functions to perform specific tasks. You should also use the PHP "include" and/or "require" directives for attaching PHP code from other files.
5. The PHP scripts must be deployed and working on tc.rnet, and the database must exist in the MySQL database on tc.rnet.

6. Anything not specified is up to your discretion. What information would you like to see in a site like this?
7. Be creative! START EARLY!!!
8. Regular backups of both code and the database are suggested, for your group's protection.

EQUIPMENT

Equipment refers to the airplanes themselves. They are identified by a unique "N" number. My particular private plane is "N9347U". They have a seating capacity (100 passengers? 80 passengers? 20 passengers?), they have crew requirements (1 pilot, 2 pilots or 3 pilots to fly? How many flight attendants?), they have a type (737, 747, 757, 767, MD80, Airbus 340, Airbus 360, RJ4, etc.).

FLIGHTS

Flights have a unique number assigned to them, are made between 2 cities, have a certain plane assigned to them (N9347U), a pilot(s) assigned to them, flight attendant(s) assigned to them, certain days they are performed (Mon/Wed/Thu/Sat) and customers make reservations with them. Flights have a price associated with them as well. CAUTION: A flight is set up in the database but the DAY they flight is made is different. Customers make RESERVATIONS based on a flight AND day! Think about it! This will be the most complicated relationship in the system.

OTHER BASIC INFORMATION

- ✓ Administrators can add, change and delete any records in the system for pilots, attendants, customers, flights and equipment.
- ✓ Administrators are solely responsible for assigning pilots and attendants to flights.
- ✓ Pilots can add, change or delete their own information.
- ✓ Flight attendants can add, change or delete their own information.
- ✓ Customers can make reservations but CANNOT change them when they have been made.

Milestone 1 **(Submission Deadline: Monday, October 17th, 2016 by 9AM.)**

1. Each team should submit an **ERD (*ERD_G##_ver1.png)*** in ***image*** format for the entire final project. This ERD should include any entities necessary for the above-mentioned tasks. Comments on your ***initial*** ERDs will be handed out in class on Wednesday, October 19th.
2. Each team should also write up a project planning document (***ProjectPlanning_G##*.txt)**. This should include information detailing which team members are responsible for which parts of the project. The delegation of work should be fairly detailed. You should also include a ***detailed*** schedule of deadlines for completion of the project to keep your group on track for the duration of the semester. I would also like to see specifics on how your team will handle communication during the project with ideally a regularly scheduled group meeting to make sure team members are keeping up with their individual tasks.

Milestone 2 **(Submission Deadline: Monday, October 24th by 9AM.)**

1. Complete ERD for the final project (with revisions if necessary).
2. A text file (**CreateTables_G##.sql)** with CREATE TABLE statements (for the MySQ6 DBMS) corresponding to your ERD.
3. Updated project planning document.

Milestone 3 **(Submission Deadline: Wednesday, November 16th 2016 by 9AM.)**

Each group should write up a group progress report that gives the current status of the project (relative to the project planning document) including:
1. Specific tasks that are complete.

2. Specific tasks that are on schedule.
3. Specific tasks that are ahead of schedule.
4. Specific tasks that are behind schedule (as well as how the group plans to catch up).
5. Any group difficulties or anticipated issues.
6. An update, detailed scheduled for ensuring the project will be completed by the deadline.

Group Meeting

Each group will meet with the instructor during the week of November 13th to discuss the group progress report. You should schedule a meeting time with me when the entire group can be there.

Demonstrations

During Finals Week, each group will schedule a time when they can meet with the instructor and TA to demo the final project. You will be asked to show us how your website meets all of the specifications listed for the project.

Deliverables

1. ***Website:*** A usable and visually pleasant web application that has all the functionality described in the above-mentioned tasks, including user authentication and web application logging.
2. Final Report: The organizational layout is below.
3. Files/DB Dump*:* A copy of all your files (code, CSS, images, etc) and a dump of your database on a CD. All file references in your code should be relative to the current directory so your application can be ported elsewhere on tc.rnet and still function. Your database should be populated with data.

Final Report

The report must follow the following organizational layout:
1. Introduction, restatement of the problem/design-goals:
a. Include links to group project site
b. Include multiple pilot credentials and at least one admin credential set
2. Delegation of project tasks. Include the members of the group and what was accomplished by each group member (be specific).
3. Data model (ERD), appropriately documented.
4. **One page (minimum) per major task**, include short description of solution and separate enumerated algorithms for PHP and SQL. This should be written by the team member that did the work. Brag about what you accomplished/learned.
5. Usage instruction/user manual. Instructions should be simple enough to allow a computer novice to learn your system.
6. Issues, frustrations, areas of difficulty. What did you collectively learn or get out of the group project? What would you do differently next time?
7. Summary

Grading Breakdown (150 points):

10 Milestone 1: Initial ERD, Project planning document
15 Milestone 2: Revised/final ERD, create table statements
15 Milestone 3: Group Progress Report
50 Functionality
30 Report
10 TA Discretion*
10 Readability/Commenting of Code
10 Proper/Efficient usage of SQL

*These points will be given based on overall design and integration of various scripts into a cohesive application and user friendliness.

Peer Evaluations

You will be asked to evaluate yourself and your group several times during the course of this project. These will be completed and submitted individually. BE SURE TO PULL YOUR OWN WEIGHT! Though grades are given on a team basis, individual grades will be adjusted appropriately for poor evaluations and/or issues with individual progress reports. These evaluations MUST be turned in to receive a grade on the final project! There are NO makeup evaluations if you miss class on that day!

Teamwork Issues

An objective of the class and specifically the group project is to develop your skills at working in teams. Keep communication lines as open as possible, as this will be essential to a successful project. You should keep your teammates aware of your progress and difficulties and ask for help when necessary. Try to resolve any group conflicts among yourselves first. If there are issues that cannot be resolved within the group, bring them to the instructor. Do not wait until near the deadline to start addressing conflicts, as it will probably too late at that point.

Summary of Due Dates

1. Milestone 1:
a. What: ERD, Project Planning document
b. Due: MONDAY, October 17th 2016 by 9AM
2. Milestone 2:
a. What: Revised ERD, CREATE TABLE statements
b. Due: MONDAY, October 24th 2016 by 9AM
3. Milestone 3:
a. What: Group progress report
b. Due: WEDNESDAY, November 16th 2016 by 9AM
4**. Final Project:**
a. What: Fully functioning website, files, DB dump, and final report.
b. Due: SUNDAY, NOVEMBER 27TH by midnight
5.Group accounts will be locked down at 11:59 PM on NOVEMBER 27th at MIDNIGHT. This means you will not have access to your code or database after this time. Be sure everything is functioning correctly before this deadline.

Start early and have fun!