2018 U.S. Information Technology Collegiate Conference Application Development Competition

Do not put your name(s) or your school's name on anything that you submit, including comments, file metadata, and images.
Doing so will result in disqualification of your team.
The only identifying information you should use is your team number.

PROBLEM STATEMENT: San Antonio Event Volunteers

Visit the following site: http://visitsanantonio.com/Browse-Book/Events/?order=Date

Your team has been tasked with developing a volunteer sign up application for the City of San Antonio yearly events. The city needs more volunteers and has developed a reward system for volunteers that help at multiple events. Your application must keep track of the rewards earned by individual volunteers. Since not all events draw the same number of participants, events are categorized into three levels: Gold, Silver, and Bronze. During testing your group can select multiple events (from the link provided above) for each level. The city will determine at a later date what events should actually be listed in which level.

All storage of information will be placed into text files, not databases. This is to remove any connection problems you might otherwise experience. You must include the text files that your application requires with your complete code. Applications which store and retrieve XML data will earn more points than applications which store information as comma (or tab) delimited.

A new volunteer must be provided the ability to register themselves in the system. The registration form should require the user to enter the following information: First Name, Middle Initial, Last Name, Userid, Password (encrypted or hashed), address, city, state, zip code, phone number and e-mail address. All fields should verify the proper format of information and that information has been entered in all fields. The verified information must be stored into a text file. An additional user level field should automatically be populated with 'volunteer' which will restrict anyone using the userid and password to only the volunteer screens.

A current volunteer can log into the system using their userid and password (Hint: Provide error checking). After logging in the volunteer will be given the opportunity to select events (Hint: use a dropdown list) in which they want to participate. When selecting the event, the event description will indicate if it is a Gold, Silver or Bronze event. They will select a four-hour time block (8-12, 12-4, 4-8) that they are available to work (Hint: Use Radio Buttons). This information, along with an empty field called 'completed' will be saved to a text file.

You will also need to create an administration screen which will allow the administrator to update the stored information to indicate when a volunteer has completed an assignment. This screen must require an administrator to log on with a userid and password that exists within the users file that includes a user level of 'administrator'. **The administrator userid and password MUST be shown in the assumptions file discussed later.**

While signed in, the volunteer should also be able to see how many points they have earned towards their rewards and what rewards they have earned. The points per category and reward levels are shown below.

Points per Level

Gold Level: 10 Points **Silver Level**: 5 Points **Bronze Level**: 3 Points

Reward Levels

- Drawing for a new car – 150 Points
- Concert or Game Tickets – 50 Points
- T-shirt or Hat – 12 Points
- Invitation to Volunteer festival – 6 points

The rewards page will only allow the user to redeem rewards for which they have enough points. For instance, if they have earned 50 points, the page should allow them to redeem everything except the "Drawing for a new car'. The volunteer can then click a redeem button which will allow the volunteer to pick the reward(s) they have selected. The point balance will be checked. If they have enough points for the rewards requested, the program will subtract the used points and store the remaining balance. The page will now inform the administrator that a request has been made by a volunteer to redeem points. The administrator will be provided with all the volunteers information and the rewards selected. How you notify the administrator is for you to decide. The volunteer should also be able to log out once he or she has completed using the application.

Requirements:

1. All logic and code must use Object-Oriented Design. Non-object-oriented submissions will be penalized 10 points compared to an object-oriented application.
2. All variables, functions, classes and objects must use meaningful names. The program name and file names must also be meaningful.
3. All data must be saved to text files only. Data stored and retrieved as XML will receive more points than regular text data. Data files must be stored and accessed from their own folder (Data).
4. Exception handling must be included. Throw exceptions for missing files and possible user entry errors. Also look out for any calculation exceptions (such as dividing by zero).

5. Menus must be provided for the user to navigate the application.
6. Strong security must be provided. Examples include: transactions, limited use of textboxes (use drop downs, lists, radio buttons, check boxes when possible). When using textboxes, check for security intrusion (such as the user trying the enter SQL or HTML information).
7. Program Documentation – Must provide internal (comments) document for the application, functions, and classes to include: IPO (Input-Process-Output) information.
8. Screen Shots: There is no required number of screen shots. However, screen shots must show ALL working parts of the application. For example (but does not include all requirements) you should have shots demonstrating a volunteer creating a userid, signing in, selecting activities to participate, selecting prizes and signing off. If a screen shot is not included for a part of the application, it is assumed that part of the application does not work. **All screen shots must be placed in one file, such as a Word or PDF document**. Reminder: Applications without screen shots will not be judged.
9. Efficient/Professional: Efficient code will be given more points than other code that accomplishes the requirements, but is not efficient. Code must be professional, well organized and designed.
10. Assumptions: As is the case when gathering requirements from users, many program details are not included in the provided description. There are cases were you must make reasonable assumptions about how the application should function. Document these assumptions in a Word, PDF, or text file and include them with your solution.

Tie Breaker (work on if you have time):

1. MVC
2. Additional Administration pages such as updates to the pages to include new prize levels and new events.

What to turn in:

Create separate subfolders (under a folder with your team name – i.e., APP_##) for each of the following. Application folder which includes all code including source code and compiled (executable) code. Data folder which includes all XML data files (with example data). Image folder which include any images used in the application. Documentation folder which includes screen shots (all in one file) and an assumptions file. Then compress all items into one ZIP file (do not use any other compression technique). Non-zip files will NOT be judged!

You MUST submit screen shots of your working code. Failure to provide screen shots will cause your application to not be judged. Even if you do not complete all requirements, you still need to include screen shots of what does work.

Code that produces errors when executed will not be judged.
You should comment out code that 'almost works'.