CS5130 Term Project Guidelines

Summary: Through this term project, students work as a team to build a full-stack web application and apply various programming techniques learned from the class(es).

Proposal Guideline

- Each team proposes project idea to the instructor until the deadline.
- Each project idea needs to have a description about the app in detail
    1. WebApp Name:
    2. WebApp Description: (what it does, what problem does it solve, etc.)
    3. WebApp Functions & Pages:
        - List the functionalities the WebApp has (create To-Do list, delete, etc.)
        - List html, js, and php files that are needed to run this WebApp
        - ex) index.html: this shows a landing page that has a login menu and explains the service this WebApp provides
        - ex) pdo.php: this contains authentication information to the database.
        - ex) xxx.js: this file handles images and color of the frontend
    4. Contribution: Explain roles each student does (stu#1: xx & xy files, stu#2: SQL, php, etc.)

Project Guideline

1. **Basic Requirement**: Each team should have HTML/JavaScript/jQuery/PHP/MySQL elements
    - HTML/CSS: CSS can be copied from somewhere (leave a note from where you copied the CSS file)
    - JavaScript: Frontend result without much of JavaScript will not get a good point
    - jQuery: should be used at least 2 times (log-in menu id validation, etc.)
    - PHP: pdo.php, $_GET, $_POST, $_SESSION, $_COOKIE
    - MySQL: all CRUD operations are needed
        - Use at least 2 tables (user login information table / product or data table)
2. **Required functions**: Each project should have these functionalities
    - Login / Logout
    - Basic securities (HTML/SQL injection)
3. **Examples**: Each project should have clear task or objective to solve. Examples will be;
    - Meeting sign-up WebApp (Apps like Zoho meeting, GoTo meeting, etc.)
        - After log-in, user can schedule a meeting with the instructor (or You)
        - Pre-scheduled meeting schedule is shown for the user.
        - When there is new meeting scheduled, it is added to the meeting schedule database.
    - Student profile management WebApp
        - After log-in, user can create new student profiles
        - Student profiles have name (First/Last), student ID, email address, etc.
        - When Student profile is made, it needs to be visualized
    - Full version Fedex tracking WebApp
        - After log-in, user can track the package.
        - Each user has multiple tracking IDs, and each user has a delivery address
        - Each package has a delivery status (In Transit / Delivered, etc.)
        - Package info can be sent through email, etc.
    - To-Do list WebApp
        - After log-in, user can add, delete, or update a To-Do list.

        - Each To-Do list is stored, retrieved, and visualized.
    - Warrensburg-MIC Bus schedule WebApp
        - After log-in, user can see today's bus schedule
        - Admin can modify bus schedule
        - If a normal user logs in, they cannot modify the schedule
    - Check out thousands of WebApps, and try to mimic the functions using languages you learned
        - User-ID generation WebApp: automatically generates cool user ID for users
        - Info-Desk ticketing WebApp: creates task to be done, delete, comments, etc.
        -

4. **Submission**: When submitting the final result, you need to submit;
    - All the files including database configuration SQL code (txt file that looks like database.txt from Assignment #2).
5. **Heads-up**
    - Multiple teams could have same topics. But codes should be completely different especially in the way it handles DB or creating frontend.
    - It takes multiple weeks to create well-made WebApp. So begin early, slowly add functions, and finish way before the deadline. If I find the same function from both side of the team, both teams will get penalty.

Presentation Guideline

1. **Time**: You'll have 5min total for the presentation
    - Presentation schedule will be announced later
2. **Process**: Please follow these procedures
    - **Slide**: Use slides to show title & description quickly (2min)
        - **Title – Description of what it does**
    - **WebApp**: Then show how the WebApp works (2min)
    - **Q&A**: instructor will ask a specific part about the code, and you should be able to explain how or why you coded in a certain way (1min)