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 (2 page, 8pts)

- 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, & DB **(5pts)**
     - List the functionalities the WebApp will have (create To-Do list, delete, etc.)
     - List html, js, and php files that are needed to run this WebApp
       - ex) index.php: 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.
     - Describe what kinds of data you'll read and write in your DB in detail
       - Ex) User firstname, lastname, email, product name, price, purchase data, user login date, etc.
  4. Contribution: Discuss thoroughly in the team, and explain roles each student does (stu#1: xx & xy files, stu#2: SQL, php, etc.) **(3pts)**

Project Guideline (50hr~100hr/team work)

1. **Basic Requirement**: Each team should have HTML/JavaScript/jQuery/PHP/MySQL elements
    - HTML/CSS, JavaScript: Use bootstrap and a separate CSS file
        - Index.php is the starting point for your project
    - Persistent input & POST-Redirect Rule is needed for POST form
    - PHP: pdo.php, $_GET, $_POST, $_SESSION, $_COOKIE
    - Cookie/Session: track how many times each user accessed the App and store it in DB
    - jQuery: should be used at least 2 times (log-in menu id validation, etc.)
    - MySQL: all CRUD operations are needed for users and products/data.
        - Use at least 3 tables (user login information table, product or data table, etc.)
            - Minimum 10 users, Minimum 10 product or data tables are needed.
            - **Admin user** and **normal user** should have different data visibility.
        - They need to have relations (Primary key, Foreign key, etc.)
        - Explain how tables were created from initial information.
        - Use Designer menu from the phpMyAdmin or any diagram to describe table relation.
2. **Required functionalities**: Each project should have these minimal functionalities.
    - Registration / Delete user / Login / Logout with session management.
    - Basic securities (HTML/SQL injection, prepare statement, input sanitization)
3. **Examples**: Each project should have clear task or objective to solve. Examples will be;
    - <u>Meeting sign-up WebApp</u> (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.
        - When user checks available timeslot, the App shows already taken time slots (reservation info not visible)
        - **Tables**: User table, date table, user visit table, etc.
    - <u>Student profile management WebApp</u>

    - 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
    - Students can enroll classes
    - **Tables**: student table, class table, user visit table, etc.
  - 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.
    - **Tables**: user table, package table, user visit table, 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.
    - **Tables**: user tables, To-do list tables, user visit tables, etc.
  - 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.
- **You need to use the brand-new PHP installation.**
- **You cannot use framework for this project.**
- **If you need, setup weekly plan (weekly sprint). This is the best for time management.**

Presentation Guideline

1. **Time**: You'll have 5~7min total for the presentation (total time will be different for each CRN)
    - Presentation schedule will be announced later
2. **Process**: Please follow these procedures
    - **Slide**: Use slides to show title & description (3~4min)
        - Max # of the slide: **8 slides** including cover page
    - **WebApp**: Then show how the WebApp works from 000webhost (1~2min)
    - **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 (1~2min)

Term Project Grading

1. **Team Registration (2pts), late -1pt**
2. **Proposal (8pt), late -2pt**
3. **Presentation (20pt)**
    - **a.** Poor (10pt) / medium (15pt) / excellent (20pt)
4. **Requirement & Functionalities (70pt)**
    - a. If your team meets all the project requirement, you'll receive 20pt.
        - i. Fair (10pt) / medium (15pt) / great (20pt)
    - b. Rest of the point (50pt) will be given based on your
        - i. App's idea level (20pt)
            1. fair (10pt) / medium (15pt) / great (20pt)
            2. rental car/restaurant/beauty shop reservation App gets 10pts
        - ii. App Functionality level (30pt):
            1. Little functionalities (10pt) / medium (20pt) / great (30pt)

5. **Example**: team submitted proposal (10) / medium presentation (15) / Meeting all the project requirement (20) with fair idea (10) + minimum functions (10), your team will get **65pts**
    a. If you worked from the files you downloaded, your team's point will range **30~50pts**
    b. **Presentation and files submission due date**
        i. **Tue class: 11/27, 11:59PM, Fri class: 11:30, 11:59PM**
        ii. **Both students need to submit the same files**