CPE 241 Database Management System

Problem Session #07

Date: check with PS07 in LEB2

Building UXUI, with queries for backend processing

Introduction

In this lab, after refining your database design with business requirements and normalization, you will focus on how administrative users interact with the system. Instead of implementing a full-scale application, we will concentrate on designing the admin interface to facilitate system maintenance and management.

You will start by defining admin user scenarios, considering how they interact with the system to perform essential tasks. Based on these scenarios, you will design the user journey, mapping out the key interactions required for smooth operation. Following this, you will create UI designs that enable admins to efficiently manage the database.

Additionally, beyond standard UX/UI considerations, you will also design admin reports that involve aggregation and time interval queries. These reports will provide valuable insights and analytics, helping admins make informed decisions based on system data.

By the end of this lab, you will have a structured understanding of admin interactions, interface designs, and reporting needs within your database system.

Objectives

By completing this lab, you will:

1. **Define Admin User Scenarios**
    - Identify how **admin users** will interact with the system for database management.
    - Specify key administrative tasks such as **data entry, updates, deletions, and report generation**.
2. **Design the User Journey (UX Design)**
    - Map out the **workflow and steps** that an admin follows to perform tasks efficiently.
    - Ensure logical and intuitive **navigation and interaction flow**.
3. **Create UI Designs for Admin Functions**
    - Develop **UI mockups** that represent key admin screens, such as:
        - **Login Page** (for secure access).
        - **Admin Dashboard** (overview of system activities).
        - **Data Management Pages** (for adding, updating, and deleting records).
        - **Reports and Analytics Section** (to generate insights from system data).
4. **Design Reports for Admin Users**
    - Identify key **admin reports** that require **aggregation and time-based queries**.
    - Create **visual representations** of these reports, such as tables, charts, and summaries.

- Examples include:
  - **User activity logs over time** (e.g., number of students registered per month).
  - **System usage statistics** (e.g., most accessed database entities).
  - **Data integrity reports** (e.g., missing or inconsistent records).

5. **Develop Backend Queries for Report Generation**
   - Write **SQL queries** to fetch, aggregate, and format data for reports.
   - The requirement for the final project will be at least three queries that join four tables with an aggregation function.
   - Ensure **efficient indexing and performance optimization** for large datasets.

Example queries include:

```
-- Count the number of students registered per department
SELECT departmentName, COUNT(studentID) AS total_students
FROM student
JOIN department ON student.departmentID = department.departmentID
GROUP BY departmentName;
-- Get the number of course enrollments per month
SELECT MONTH(registration_date) AS month, COUNT(enrollmentID) AS total_enrollments
FROM enrollment
GROUP BY month;
```

6. **Build a Clickable Prototype of Your System**
   - Use prototyping tools such as **Figma, Adobe XD, or Balsamiq** to create an interactive admin interface.
   - The prototype should simulate:
     - **Admin login and navigation**.
     - **Managing database records** (adding/updating/deleting users, courses, etc.).
     - **Generating and viewing reports**.

Steps to Complete the Lab

Step 1: Identify Admin Use Cases

- List down specific admin **tasks and workflows**.
- Define user roles and **access control** (e.g., super admin vs. data entry admin).

Step 2: Create UX Flow (User Journey)

- Develop a **flowchart or diagram** showing the steps an admin takes to complete tasks.
- Consider **error handling and security measures** in navigation.

Step 3: Design UI Mockups

- Sketch or create **wireframes** for each important admin interface.
- Ensure **consistency in UI elements**, such as buttons, forms, and navigation bars.

Step 4: Design Admin Reports

- Determine **what insights are useful for admins**.
- Mockup **report pages** with **tables, charts, and filters** for time-based analysis.

Step 5: Implement SQL Queries for Reports

- Write and test **queries for data aggregation and reporting**.
- Optimize queries to handle **large datasets efficiently**.

Step 6: Build a Clickable Prototype

- Use tools like **Figma** to create an **interactive simulation of the admin UI**. (optoinal)
- Ensure it allows navigation and demonstrates key interactions.

In this step, you will create a **clickable prototype** of your admin system using **HTML**. Instead of using prototyping tools like Figma, you will build a **basic web-based prototype** with multiple pages linked using the `<a href="">` tag. This will allow you to navigate between different sections of the admin system, simulating how an admin would interact with the system.

**Here are some HTML tutorials for you.
w3schools
geeksforgeeks
tutorialspoint

Example: How to Build a Clickable Prototype

This is an example showing you how to create a top level long in page link to the menus and you will have to add more pages of your designs.

1. **Create Separate HTML Pages for Each Admin Section**
   You will need to create different HTML files, each representing a key admin page.
   Example pages:
   - `index.html` → Home Page (Dashboard)
   - `login.html` → Login Page
   - `manage_users.html` → User Management Page
   - `manage_courses.html` → Course Management Page
   - `reports.html` → Reports Page

2. **Use the `<a href="">` Tag to Link Pages**
   The `<a>` tag is used to navigate between different pages.

Example of linking pages using **HREF**:

```
<a href="manage_users.html">Manage Users</a>
<a href="manage_courses.html">Manage Courses</a>
<a href="reports.html">View Reports</a>
```

3. **Create a Simple Navigation Menu for Each Page**
    - To make navigation easier, **each page** should have a **menu** linking to all other sections.

Example navigation menu (to be included in every page):

```
<nav>
    <a href="index.html">Dashboard</a> |
    <a href="manage_users.html">Manage Users</a> |
    <a href="manage_courses.html">Manage Courses</a> |
    <a href="reports.html">View Reports</a> |
    <a href="login.html">Logout</a>
</nav>
```

4. **Example of Creating and Linking Pages**

`index.html` (Dashboard Page)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Admin Dashboard</title>
</head>
<body>
    <h1>Admin Dashboard</h1>
    <nav>
        <a href="manage_users.html">Manage Users</a> |
        <a href="manage_courses.html">Manage Courses</a> |
        <a href="reports.html">View Reports</a> |
        <a href="login.html">Logout</a>
    </nav>
    <p>Welcome to the admin dashboard. Select an option above.</p>
</body>
</html>
```

`manage_users.html` (User Management Page)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Manage Users</title>
</head>
<body>
```

```
    <h1>Manage Users</h1>
    <nav>
        <a href="index.html">Dashboard</a> |
        <a href="manage_courses.html">Manage Courses</a> |
        <a href="reports.html">View Reports</a> |
        <a href="login.html">Logout</a>
    </nav>
    <p>Here you can add, edit, or delete users.</p>
</body>
</html>
```

`reports.html` (Reports Page)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Reports</title>
</head>
<body>
    <h1>Admin Reports</h1>
    <nav>
        <a href="index.html">Dashboard</a> |
        <a href="manage_users.html">Manage Users</a> |
        <a href="manage_courses.html">Manage Courses</a> |
        <a href="login.html">Logout</a>
    </nav>
    <p>This page will display reports based on database queries.</p>
</body>
</html>
```

5. **Test Your Prototype**
    - Open **`index.html`** in a browser.
    - Click the links to navigate between pages.
    - Make sure every link correctly **redirects** to the intended page.

Deliverables

1. **Admin User Journey Diagram** (UX flow).
2. **UI Mockups** for admin pages.
3. **Designed Reports** with sample data.

4. **Backend SQL Queries** for generating reports.
5. **Clickable Prototype** demonstrating admin functionalities of your design.

Submission

Combine results from all deliverables in a PDF file then submit the report to PS07 on LEB2 before the deadline.