ITCS333: Internet Software Development - Course Project Specification

Project Overview

Welcome to the ITCS333 course project! The goal of this project is to apply the concepts learned throughout the semester to build a real-world, dynamic web application. You will work in teams of five to create a fully functional "Course Page" website from the ground up.

This project will challenge you to design, develop, and deploy a full-stack application, covering everything from front-end layout with HTML/CSS to back-end logic and database interaction with PHP and MySQL.

Learning Objectives

Upon successful completion of this project, you will be able to:

- Design and structure a multi-page website using **HTML** and **CSS**.
- Enhance user interfaces with client-side **JavaScript**.
- Develop a server-side application using **PHP**.
- Design and interact with a **MySQL** database using **PDO** for secure data

persistence.
- Implement full **CRUD** (Create, Read, Update, Delete) functionality

Technical Requirements

- **Front-End**: HTML, CSS, JavaScript.
  - You are permitted to use a CSS framework like **Pico.css** or **Bootstrap** to streamline your design process.
- **Back-End**: PHP.
- **Database**: MySQL, with connections handled exclusively via **PDO**.
- **Version Control**: Git & GitHub Classroom.
- **Hosting**: Replit.

Team Structure & Core Responsibilities

The project is designed for teams of **five students**. Each student is the **primary owner** of one of the five tasks described below. While you own a specific task, you are expected to collaborate with your team to ensure the entire application is cohesive and functional.

Crucially, the student assigned to **Task 1** is responsible for creating the central authentication (login/logout) system. All other team members **must** integrate with and use this system to protect their respective pages.

Task Breakdown

Task 1: Homepage, Admin Portal & User Management

Owner: Student 1

This is the foundational task. It includes the public-facing homepage and the core administrative backend for managing users.

Features:

- **Homepage**: A simple, elegant landing page for the course.
- **Login System**:
  - A login page for all users (teachers and students).
  - A secure logout mechanism.
  - PHP session management to track logged-in users.
- **Admin Portal**: Accessible only to the Teacher/Admin user.
  - **Password Management**: The admin must be able to change their own password.
  - **Student Management (Full CRUD)**:
    - **Create**: Add new students to the course (e.g., with a name, student ID, email, and a default password).
    - **Read**: List all registered students in a clear, tabular format.
    - **Update**: Edit a student's information.
    - **Delete**: Remove a student from the course.

Task 2: Course Resources

Owner: Student 2

This section is for sharing course materials.

Features:

- **Admin View (Full CRUD)**:
  - The teacher can add new resources (e.g., book chapters, lecture notes, web links), each with a title and description.
  - The teacher can edit or delete existing resources.
- **Student View (Read-Only)**:
  - Students can view a list of all resources.
  - Clicking on a resource takes them to a dedicated page for that item.
- **Discussion**: On each resource's dedicated page, logged-in students and the teacher can leave comments to discuss the material.

Task 3: Weekly Breakdown

Owner: Student 3

This section organizes the course content on a week-by-week basis.

Features:

- **Admin View (Full CRUD)**:
  - The teacher can add a new entry for each week of the semester (e.g., "Week 1: Introduction to HTML").
  - Each weekly entry can contain a description, notes, and links to exercises or resources.
  - The teacher can edit or delete existing weekly entries.
- **Student View (Read-Only)**:
  - Students can see the breakdown for all weeks.
  - Clicking on a week opens a detailed view.
- **Discussion**: In the detailed view for each week, logged-in students and the teacher can post comments or ask questions.

Task 4: Assignments

Owner: Student 4

This page lists all course assignments.

Features:

- **Admin View (Full CRUD)**:
  - The teacher can add new assignments, including a title, description, due date, and attached files (or links to them).
  - The teacher can edit or delete existing assignments.
- **Student View (Read-Only)**:
  - Students can view a list of all assignments and their details.
  - Clicking an assignment opens a detailed view.
- **Discussion**: On the detailed assignment page, logged-in students can ask clarifying questions, and the teacher can respond.

Task 5: General Discussion Boards

Owner: Student 5

A general forum for course-related discussions not tied to a specific resource or assignment.

Features:

- **All Users (Teacher & Students)**:

    - **Create Topics**: Any logged-in user can create a new discussion topic with a subject and a message.
    - **Read Topics**: All users can view the list of topics.
    - **Update/Delete (Ownership-based)**: Users can only edit or delete the topics or comments that they themselves have created. The teacher (Admin) has the ability to delete any topic or comment.
- **Commenting**: All logged-in users can reply to any existing topic.

Development Phases & Final Deadline

The project will be developed in three distinct phases. Your final submission must be a complete, fully functional application.

Phase	Title	Description
1	**HTML/CSS Mockups**	Create the static HTML and CSS files for all pages in your task. The pages should be well-structured and styled but will not have any functionality.
2	**JavaScript Interactivity**	Enhance the front-end with client-side JavaScript. This includes form validation, dynamic UI elements, and preparing for back-end integration (e.g., using `fetch` for AJAX calls).
3	**PHP/MySQL Backend**	Implement all server-side logic. Connect to the MySQL database using PDO, implement CRUD operations, and manage user sessions. This phase brings the application to life.

Final Project Deadline: Saturday, November 29, 2025

Submission Guidelines

- Your team will be given a template repository via **GitHub Classroom**.
- All work must be committed regularly to this repository. We will be checking the commit history to assess team collaboration.
- The final submission is the link to your team's completed GitHub repository.
- Your repository **must** include a `README.md` file that clearly lists:

- The project title.
- The names and student IDs of all team members.
- A table indicating which student was responsible for which task.
- A link to your live, hosted application on Replit.

Happy Hacking!