# **Task3 Documentation**

VXD240002, Dacharla Venkata Abhiram

## Introduction

In this task, I enhanced the Course Management System by adding advanced CSS, responsive design, and interactive features. The system now includes a header, footer, and navigation bar, along with functionalities like adding new courses, searching for courses, and displaying course details dynamically. Below, I explain how I implemented each requirement step-by-step.

## File Organization

The project files are organized as follows:

* **task3.php**: The main PHP file from task2 (task2.php) with Bootstrap integration.
* **task3.js**: JavaScript file containing test cases from task2 and additional functions.
* **task3.css**: Custom CSS file for styling the web page.

A computer screen with a white background

AI-generated content may be incorrect.

## Step-by-Step Implementation

## 1. Header, Footer, and Navigation

## I created a header with the title "Course Management" and a footer with copyright information. The navigation bar includes links to Home and Add Course. I used Bootstrap for styling and ensured the header is fixed at the top of the page for easy navigation.

## For this

* I added a <header> tag with a background color and centered text.
* I used Bootstrap's navbar component for the navigation bar.
* I made the header static using position: fixed in CSS so it stays visible while scrolling.
* I added a footer with links to Privacy Policy and Terms of Service.

Header code with styles

A screenshot of a computer

AI-generated content may be incorrect.

Footer code with styles

A screenshot of a computer screen

AI-generated content may be incorrect.

## 2. Display Course Information

Initially, each course is listed in a short form with the course/section number and title. When a user clicks on a course, the full details (e.g., instructor, schedule, location) are displayed dynamically. For this

* I created a div for each course with a toggle button (▼) to expand/collapse the details.
* I used JavaScript to handle the click event and toggle the visibility of the course details.
* The course details are fetched from the database and displayed in a structured format.

A screenshot of a computer screen

AI-generated content may be incorrect.

JavaScript part for toggling event

A screenshot of a computer screen

AI-generated content may be incorrect.

As requested initially each course will be listed in a short form with course/section number and title. By clicking on it, the full course information will be displayed.

A screenshot of a computer

AI-generated content may be incorrect.

When clicked on it it pops out with full information that is in our table.

A screenshot of a computer

AI-generated content may be incorrect.

## 3. Add a New Course

I added a button to open a modal form for adding a new course. The form includes fields for course details like course number, title, instructor, schedule, and location.

* I used Bootstrap's modal component to create a pop-up form.
* The form includes input fields for all necessary course details.
* When the form is submitted, the data is sent to the server using PHP and added to the database.

A screenshot of a computer

AI-generated content may be incorrect.

You can see AddCourse option at the top of the page in nav bar and when clicked it opens a form which when submitted adds a new course.

A screenshot of a computer

AI-generated content may be incorrect.

When clicked on submit, the requested course is added to the table.

A screenshot of a computer

AI-generated content may be incorrect.

## 4. Search Functionality

I implemented a search bar that allows users to search for courses by course number, title, or instructor name. The search results are displayed dynamically as the user types. For this

* I added a search input field at the top of the page.
* I used JavaScript to filter the course list based on the search query.
* The search functionality highlights matching courses and scrolls to them for easy navigation.

A screenshot of a computer screen

AI-generated content may be incorrect.

Here is its appearance in the webpage

A screenshot of a computer

AI-generated content may be incorrect.

## 5. Responsive Design

I ensured the web page is fully responsive and works well on mobile devices. The layout adjusts dynamically based on the screen size. For this

* I used Bootstrap's grid system and responsive utilities.
* I added media queries in the CSS to adjust the layout for smaller screens.
* The navigation bar collapses into a hamburger menu on mobile devices.

Here are the media querries I used to make the webpage responsive.

A screenshot of a computer screen

AI-generated content may be incorrect.

Output in 691x668 measurements.

A computer screen shot of a computer screen

AI-generated content may be incorrect.

Output in Iphone XR

A screenshot of a computer

AI-generated content may be incorrect.

**6. Add Test Case Buttons**

I added buttons for Add, Update, and Delete test cases. These buttons are grouped into frames, with each frame containing the three buttons for a specific test case.

* I created a container for the test case buttons using Flexbox.
* Each frame contains three buttons: Add, Update, and Delete.
* I styled the buttons with a consistent color scheme and hover effects.
* The buttons are arranged in a single row on desktop screens and stacked vertically on mobile screen.

A screenshot of a computer screen

AI-generated content may be incorrect.

Appearance of this Test Cases in different devices.

A computer screen shot of a computer screen

AI-generated content may be incorrect.

A computer screen shot of a computer program

AI-generated content may be incorrect. A screenshot of a computer

AI-generated content may be incorrect.

**Functionality of these Testcases:**

* In this task, I enhanced the test case buttons from Task 2 to make them visually appealing and more organized.
* The functionality remains the same: Add Test Case 1 adds a new course, Update Test Case 1 modifies it, and Delete Test Case 1 removes it. This pattern repeats for Test Cases 2 through 5.
* To improve usability, I grouped the Add, Update, and Delete buttons for each test case into separate frames, arranging them side by side.
* This makes it easier to perform all actions for a specific test case in one place. I also styled the buttons consistently and added hover effects for better interactivity.

When Add Test Case 1 is clicked it opens add course form with pre loaded course details which are written in task3.js

A screenshot of a computer

AI-generated content may be incorrect.

When submitted it displays in the table.

A screenshot of a computer

AI-generated content may be incorrect.

When Update Test Case1 is clicked it directly updates the course field with pre-set values.

A screenshot of a computer

AI-generated content may be incorrect.

When Delete Test Case 1 is clicked it directy deletes that course.

A screenshot of a computer

AI-generated content may be incorrect.

Similarly it goes for all other buttons too.

**Conclusion**

In this task, I successfully implemented a responsive and interactive Course Management System. The system includes a fixed header, dynamic course details, a modal form for adding new courses, and a search functionality. I used Bootstrap for styling and ensured the web page is fully responsive. Screenshots of the code and outputs are included in the document for reference.