Project #4

CIS 4510
Oakland University

Note: ***You may work on this final project in a group of up to 3 people if you would like. It may also be done individually. If you work in a group, you must submit the deliverables, and include ALL group member names and what each contributed, specifically in both the comment when submitting as well as in the report.***

Make this a solid portfolio-ready project that you would be happy to show a future employer.

Objectives

- Design and implement a full-stack web application using modern frontend and backend technologies.
- Demonstrate proficiency in HTML, CSS, and JavaScript to create a responsive and user-friendly interface.
- Integrate a backend server using either PHP with MySQL or Node.js with MongoDB to handle data operations.
- Consume and process data from third-party APIs and dynamically incorporate that data into their application.
- Store and manage API data in a custom database, applying appropriate schema design and data transformation.
- Develop a logical database schema that supports persistent user or application data.
- Utilize RESTful principles to manage client-server communication.
- Practice good development habits including code organization, modularity, and separation of concerns.
- Produce professional-quality project documentation, including a usage guide, database design summary, and screenshots.
- Demonstrate creativity and problem-solving by proposing and executing an original project idea.

Requirements

For your final project, you will **come up with an idea of your own**, **and then** design **and** implement a complete full-stack web application. This is your opportunity to show what you've learned and build something that you're personally interested in.

The only hard requirements are that your app must:

- Use HTML, CSS, and JavaScript (React is encouraged but not required)
- Include a backend:
    - Either PHP with MySQL
    - Or Node.js with MongoDB
- Incorporate at least one third-party API
    - E.g., download certain data from an API (or more than one) and store it in your own MySQL or MongoDB database.
- Save some API data to your own database
- Be visually appealing and responsive

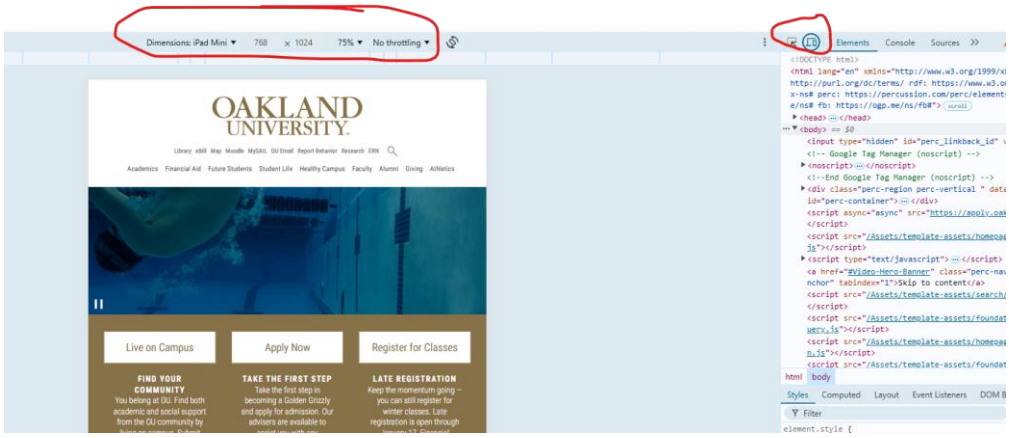
Project ideas

You may come up with your own idea, or be inspired by one of the following:

- **Weather Tracker**
    - Pull data from a weather API, allow users to save favorite locations, and see historical trends saved in a database.
- **Book or Game Collection Manager**
    - Use a public API (like Google Books, Open Library, or RAWG for games), let users save entries to a personal collection, add notes, and view saved data from your own DB.
- **Recipe Finder + Grocery List**
    - Use a recipe API, save recipes, and generate a shopping list stored in the DB.
- **Movie/TV Tracker 2.0**
    - Build on Project 2, but now save favorites in a real database, include user comments or ratings, and expand the UI.
- **Event or Meetup Finder**
    - Use a public events API and let users save events they're attending to a local DB.

Deliverables

- Include **screenshots** of your Webpage working in desktop, tablet, and mobile sizes at the top level of the folder of your projects
    - This can be done by either manually changing the size each time, or by using the tools built into the browser, such as by using **Chrome Developer Tools (F12 tools)**
- Include a report with screenshots, **how you designed your database and Website**, and a "How-To" guide



- Zip your entire project (along with screenshots) containing:
    - Report with screenshots and How-To Guide
    - Frontend code
    - Backend code
- Upload the entire zip file to Moodle on or before the due date.