

# MET CS401/MET CS601: Term Project



## Instructions

Create a website that uses the API's and technologies we discussed during the semester.

## Grading

See the sections below in this document for details.

## Browser Requirements

The latest releases for any of these: Chrome, Edge, Firefox and Safari.

## Mobile Device Requirements

Assume a device that has been release in the past 2 years.

## High-level Requirements

1. The student can host their project on a free public hosting service or on localhost. There will be a loss of 10 points from the overall grade if the student simply uses the file:// protocol.
2. Students should develop/design their work to be a mobile first design, then desktop with media queries and/or responsive design patterns
3. There shouldn't be any in the browser console. The point loss you receive on this point depends on the number of errors in the console.
4. Must use CSS extensively. Do not use HTML tables to structure your website's pages. If you do, you will lose 10 points immediately upon grading.
5. You cannot cut and paste the same CSS rules over and over again in each HTML page to meet the minimum number of selectors used. (See CSS requirements)
6. Should not copy javascript from other sources and say you did it. You can learn from them. Integrate with them, but do not simply take their work, put into your page and say you developed it.
7. You do not need to have a database, but you can use one if you want. That's completely optional and up to you. We don't have any points to give for your extra effort on creating

or using a database. However, it is GOOD experience and you should try if you have time.

8. The minimum HTML pages you need to have: 1 (one). That is the bare minimum for a SPA webapp. However, you need to have or show a complete website with content and purpose. In general, you should not need to create more than 10 views/pages. (Having a high page count does not mean a good grade)

## Previous Ideas

In the past, students have created the following types of websites:

- E-Commerce site
- Hangman Game
- Personal Diary
- Restaurant Ordering System
- Dating app
- Workout notes
- Weather app
- Rewrite of the StudentLink
- Todo/Reminder app
- Personal Bio website
- Healthcare website
- Catering System

## Look and Feel

Your website should be visually appealing. You should use images, good CSS and purposeful javascript that does useful things. <https://bit.ly/37rykNV>

## CSS Requirements

- a. Must use at least 10 descendent selectors.
- b. Must use at least 10 unique adjacent selectors.
- c. Must have 2+ different class selectors.
- d. Must have at least 2 ID selectors.
- e. Must have 5 or more pseudo element selectors.
- f. Must have 5 or more pseudo class selectors

## Javascript Requirements

- a. Use of React, Vue or Angular is fine. But, you do not have to use these frameworks.
  - i. Javascript code should use ES6 modules. You don't have to - but you should.
  - ii. You must have at least 20 active event handlers. They can be of any type. (click, select, drop/drag, mouse events, window events).
  - iii. Must have at least one functional and in-use ES6 class in your active codebase. Meaning, you can just declare a ES6 class and not use it.
  - iv. Global functions are generally not a good idea. If you need to declare a Globally-scoped function, you need to put in a comment on top of it explaining why. Please use Javascript Modules instead.

- v. You need to show proficiency using functions. Do not simply cut paste the same functions over and over again. Your functions should be unique and do something useful to your web application.

Since you will have at least 20 active event handlers from the previous requirement, you should have enough coded to meet this requirement. But, you need to show a mix of:

- Function Expressions
- Function Declarations
- Arrow Functions

- b. The website must use Ajax in some way. The minimum number of Ajax calls in your pages/views are 4 CRUD operations actively used. (You can use Axios, Fetch or native XHR)

## Optional Requirements

1. Students can use ReactJS, Vue or Angular if they choose to.
2. Students can create a backend system with NodeJS, PHP, Java, etc.. **It is not required.**
3. Students can choose to develop their website as a SPA (Single Page Application)
4. Students can use any open source component. However, student must show their own work and Javascript abilities and not rely on using open-source for the entirety of their project.
5. Students can develop a database. **It is not required.**



*© 2021. The Professor reserves the right to update this document.*