Overview

The goal of this final exam project is to practice concepts covered in class by implementing a Node.js application. What the application does is up to you. For this work you can work with classmates (up to four persons per group) or by yourself. There is NO LATE DEADLINE for this work. The project is due on Monday, May 19, 12:30 pm (NOTICE IT IS NOT THE USUAL TIME). Please do not ask whether your group can have more than four people; the maximum will be four.

Warning: Every semester I have some students that contact me after the deadline indicating they thought the project was due later in the day (as it is the case for other projects). Please avoid this situation. I will not accept any work after the deadline. Remember, there is NO late deadline. If you forgot to submit, were late a little bit, had network problems, etc. and that made you miss the deadline, then I may accept your work, but I will apply a 50 pts penalty. Plan to submit your work the day before it is due.

Requirements

1. Design and implement a Node.js/Express.js/MongoDB application.
2. Your application must rely on Node.js and Express; no PHP code will be accepted.
3. You must use routes (express express.Router()) for some of your endpoints.
4. Your application must store and retrieve data from MongoDB.
5. Your application must have at least one form where users provide some data.
6. Your application must define and use a CSS file that:
   - Relies on at least the background-color, color and font-size properties
   - Relies on at least a google font
7. Your application must rely on at least one API that provides some data (weather, pictures, etc.) for you. Some API resources (you are not restricted to only these resources) are:
   - https://rapidapi.com/
   - https://developer.vonage.com/blog/2021/03/15/the-ultimate-list-of-fun-apis-for-your-next-coding-project
   - https://apipheny.io/free-api/
   - https://github.com/public-apis/public-apis
8. Using setInterval, setTimeout, fetch by themselves (without downloading data from somewhere online) will not satisfy the API requirement. If you plan to use an API I have used in class examples or projects, you need to contact me first so I can approve it.
9. You can use any libraries (e.g. React), frameworks, tools (e.g., Vite) you want as long as you satisfy the requirements specified above.

Grading

1. (45%) Use of Node.js/Express/express.Router().
2. (14%) Application stores and retrieves data from MongoDB.
3. (8%) Application relies on at least one form.
4. (8%) CSS file requirement.
5. (20%) API requirement
6. (5%) Application is deployed online (e.g., using a company that provides a free-hosting option (e.g., https://render.com/).

Additional Information

1. Your application can be as simple as you want. You will get full credit as long as you satisfy the above requirements.
2. The complexity of your application will not depend on the number of people in your group.
3. Feel free to help classmates (even if they are not part of your group).
4. Applications that are extensions of class projects will not receive any credit.
5. You must have a text file called README.md in the main directory of your application. This file will have the information specified below.
6. Please, check the clarifications Piazza folder associated with this work often.
7. Feel free to use the Piazza final folder to find classmates for a group.
8. You should use this project as an opportunity to start developing an app that eventually you could show potential employers. Implement the app so it satisfies the project requirements, and after the class is over, continue expanding it.
9. We may start grading this project the last day of classes for those that have finished and would like to know their score. One advantage for you is there are any problems with your project, you can fix them and we will regrade. For us, it will help us finish the grading sooner :). Additional information about the grading process will be posted in Piazza.

README.md file

You must have a text file called README.md in the main directory of your application with the following information:

1. Who Submitted the Project in the Submit Server - Name and directory id of the group member that uploaded the application to the submit server.
2. Group Members - Add the name and directory id (in parentheses) of each member of your team. If you are working by yourself, add your name and directory id too.
3. App Description - One or two line description.

4. YouTube Video Demo Link - Link to a YouTube video that provides a demo of your application. This video is very important. We will not grade a project unless a YouTube video is provided. The video does not need to have audio (narration).
5. APIs Information - Names and link to API(s) used by your application.
6. Contact Email - Which group member should we contact if we have questions.
7. Deployed App Link - Link to the online deployment of the app (e.g., link to Render entry).

Below we included an example of the information we are expecting. Please follow this format. Please use markdown (https://markdowntohtml.com/) so we can select links you provide in the information.

```
           Submitted by: Rose Smith (directory id: superterp)
            Group Members: Rose Smith (superterp), Peter Santos (pmegaterp)
            App Description: Allow users to find images, songs of an author and
store the results.
            YouTube Video Link: https://www.youtube.com/MyAmazingAPP
            APIs: VideoApp (http:notrealvideoapp.what), SoundApp
(http:notreadlsoundapp.what)
            Contact Email: superterp@notreal.address.what
            Deployed App Link: https://notreal.what.terpNel/myApp
```

Submission

1. PLEASE, ONLY ONE SUBMISSION PER GROUP.
2. Remove the node_modules folder from your application. Make sure we can recreate this folder when executing npm install.
3. Make sure the README.md file described above is part of the project. It should appear in the main directory of your application.
4. Upload a zip file with your work using the submit server entry "Final Exam Project."
5. Make sure that after uploading your work, you download the work and verify you are not missing any files. If you are missing any files, you will be penalized -12 pts.
6. We will only grade the last submission you provide.
7.