Guidelines for Web Programming Project

Tasks to do during implementation

After you've turned in your Project Proposal, you are now ready to begin implementation!

We suggest the following tasks for your project. Each task must be completed and evaluated within the given time.

Evaluation Date	Task title	Description
Task 1 **(25/03/2022)**	Implement the frontend UI with minimal-to-no CSS	**Create a class diagram for your frontend JavaScript** • Before you begin any coding, write a class diagram for your frontend **Implement one screen at a time, without CSS** • Write the HTML and JavaScript necessary to implement the behaviour of one of the screens in your app. Implement one screen at a time. After each screen works individually, implement the interactions between the screens. • Don't write your CSS yet, or at least don't spend too much time on CSS. You can make your web app prettier after you've gotten your web app working end-to-end. **Use fake data instead of fetch()** • In the areas where you are supposed to call fetch(), fake out the data that you could get from the fetch() call. You can fake the data by creating a hard-coded JavaScript object literal and using that as if you had retrieved that data from fetch().
Task 2 **(1/04/2022)**	Design your database	**Decide how you are going to store your data** • For MongoDB backend: ○ What collection(s) are you going to have? ○ What fields are you going to store in each object? **Write a non-server NodeJS test script to query your database** • Before you touch your NodeJS server code, write a separate NodeJS script that

		queries and updates your database. This way you can make sure you know how to make the right calls to MongoDB, independent of your frontend or server code.
Task 3 **(8/04/2022)**	Implement the backend	**Move your frontend code to public/** • Move the frontend code you wrote to the public/ directory of your backend, if it’s not there already • Verify the frontend code is served statically from public when you run your NodeJS server **Write “stubs” for your routes without involving data yet** • Write stubs for all of your GET and POST handler(s), where you only do the following: o Specify the path for the route o Print out the parameters received • Don’t query/update your database yet **Connect your frontend to your backend** • Modify your frontend code to query your server via fetch() with the appropriate parameters • Make sure the backend prints the correct parameters when queried from the frontend. o Use console.log() statements on both the frontend and backend to debug **Implement your routes** • Now implement the rest of your stubbed routes, querying or updating your database as necessary.
Task 4 **(15/04/2022)**	Finish your CSS	• After your website is working end-to-end, finish styling your frontend with CSS.
Task 5 **(22/04/2022)**	Deploy and test	• Re-deploy your website onto Heroku. • Test out your deployed application to make sure it still works as expected. • Go through the Project Requirements and make sure your app fulfills all the ones we’ve asked for.