

## 1.1. Background & Motivation

This assignment is the process you building the front-end for that MVP (Minimum viable product) to the standards described. Features need to be implemented in order for your React.js app to meet the requirements of the task, and to operate with the backend described in the spec.

The requirements describe a series of **screens**. Screens can be presented as popups/modals, or full-page views. The term *Screen* is used to refer to a specific state or representation of the web application. You have the flexibility to decide how to implement them, whether as modals, separate pages, or other appropriate UI components.

## 1.3. Setup & Requirements

### 1.3.0 IMPORTANT setup

You will **.htmlfrontend** folder. Failure to align with this requirement will result in a 50% penalty of the mark.

2. We strongly recommend you to use **nvm** to manage node version, once you have **nvm** installed, you can run **nvm use** in frontend directory to switch to **v20.17.0**, and note that for the backend we are using a different node version specified in a different **.nvmrc**.

### 1.3.1 Languages

- You can use any UI libraries that you would like, such as material-UI, Radix UI or any other library that you would like.
- Besides those described to avoid, you can install and use any package that is available on [npm](#).  
You **MUST** commit **package.json** changes.
- You may use small amounts of general purpose code (not specific to the assignment) obtained from a site such as Stack Overflow or other publically available resources. You should attribute **clearly** the source of this code in a comment with it. You can not otherwise use code written by another person.

### 1.3.2 Browser Compatibility

- You should ensure that your web app has been tested on Google Chrome (various operating systems) – make sure it's latest version.

#### 1. 3. 3 Using code found online

- You may use small amounts of general purpose code (not specific to the assignment) obtained from a site such as Stack Overflow or other publically available resources. You should attribute **clearly** the source of this code in a comment with it. You can not otherwise use code written by another person.

#### 1. 3. 4. ! Prohibited Usage !

- You are strictly NOT allowed to use universal css for styling. You cannot use direct CSS importing. You should either use CSS Modules, Styled Components, a UI Framework, or a combination of these. If you use a CSS Framework, it's fine to have ONLY setup CSS files for Tailwind CSS, any UI library you chose to work with and global setup to overwrite **body**'s style.
- You cannot use other declarative frameworks, such as Angular, or Vue. You must use ReactJS.
- You must use React.js solutions wherever possible, and avoid doing any direct DOM manipulation unless completely unavoidable (check with course staff).

## 3. Analysing the pages

### 3. 1. Analysing the Pages

Two things will want to seek external help for are:

### 3. 2. Font Sizes

Navigate to the `frontend` folder and run `npm install` in that folder to install the ReactJS app. Then run `npm run dev` to start the ReactJS app.

Generally the best approach is to set a basic font size (e.g. [React.js](#)), see how it looks, and if it just generally seems too big or too small, then adjust the

Don't forget to put this code in the head of each webpage you make:

#### 4.1. Correct viewing settings

You are **PROHIBITED** from modifying the backend unless it's specifically required in deployment.md (in your repo). No extra work needs to be done on the backend. It's provided to you simply to power your frontend.

The backend server exists in your individual repository. After you clone this repo, you will note that there's a `.nvmrc` file inside the backend directory, it's strongly recommended to run `nvm use` to use the specified node.js version to avoid potential issues for running the server, then you must run `npm install` in `backend` directory once.

To run the backend server, simply run `nvm use` each time you reopen your repo, then `npm start` in the `backend` directory. This will start the backend.

You are restricted from using any third party CSS libraries when completing this assessment. Basically, this means you can't import code using the `http://localhost:5005` and

Your backend is persistent in terms of data storage. That means the data will remain even after your express server process stops running. If you want to reset the data in the backend to the original starting state, you can run `npm run reset` in the backend directory. If you want to make a copy of the backend data (e.g. for a backup) then simply copy `database.json`. If you want to start with an empty database, you can run `npm run clear` in the backend directory.

You can also run `npm run test` in the backend directory to run the tests, the tests for `store` will give you ideas on how to structure your `store` object.

Once the backend has started, you can view the API documentation by navigating to `http://localhost:[port]` in a web browser.

The port that the backend runs on (and that the frontend can use) is specified in `frontend/src/backend.config.json`. You can change the port in this file. This file exists so that your frontend knows what port to use when talking to the backend.

Please note: You **CANNOT** modify the backend for bonus marks.

## 4. Constraints & Assumptions

- The specification is intentionally vague to allow you to build frontend components however you think are visually appropriate. Their size, positioning, colour, layout, is in virtually all cases completely up to you. We require some basic criteria, but it's mainly dictating elements and behaviour.
- For specified tasks, if applicable, pseudo-class behaviour satisfies the task requirements

## 5. Marking Criteria

The work you submit must be your own work. Submission of work partially or completely derived from any other person or jointly written with any other person is not permitted.

The penalties for such an offence may include negative marks, automatic failure of the course and possibly other academic discipline. Assignment submissions will be examined both automatically and manually for such submissions.

Relevant scholarship authorities will be informed if students holding scholarships are involved in an incident of plagiarism or other misconduct.

**Please note:** Your contributions will be measured based on the lines and commits contributed via gitlab. Please commit via your own machine or account. If you're in a pair, your contributions will not be considered yours if it is your partner who pushes the code to gitlab.

**Please note:** When special consideration is granted for one individual in a pair, it will only either 1) extend the deadline for the person who gets special consideration (it does not extend for the other individual); or 2) Result in a scale of the mark. To determine which outcome is appropriate, the person who receives

special consideration is required to email the lecturer to notify them of how the work is split up prior to deadline.

## 6. Git Commit Requirements

Every time you make commits or pushes on this repository, you are acknowledging that the work you submit is your own work (as described above).

### 4.2. Fair use of methods

- Features implemented that satisfy requirements as outlined in section 2.
- You **MUST** update the `progress.csv` file in the root folder of this repository as you complete things partially or fully. The valid values are "NO", "PARTIAL", and "YES". Updating this is necessary so that your tutor knows what to focus on and what to avoid - giving them the best understanding of your work and provide you with marks you have earned. Failure to correctly fill in this file will result in a 5% penalty.

### 4.3. Browser Compatibility

- Features implemented in a mobile responsive way that work on screens as small as 400px wide, 700px high
- Responsive design will contribute up to one quarter of the marks of this section

### 4.4. External libraries

- Your application is usable and easy to navigate. No obvious usability issues or confusing layouts/flows.
- Your application makes intelligent use of UI/UX principles and patterns discussed in the UI/UX lectures.
- Describe any attempts you've made to improve the UI/UX in **UIUX.md**. This section will ONLY be marked on things described in that file.

### 5.1. Visual Compliance (50%)

- Your code is clean, well commented, with well-named variables, and well laid out as highlighted in the course style guide.
- Code follows common ReactJS patterns that have been discussed in lectures and as highlighted in the course style guide.

## 5. 2. Code Quality (50%)

- Submitted code is completely **eslint** compliant based on provided eslint configuration file. There are no partial marks.

## 6. 6. Testing (5%)

- 60% of the marks (3/5) received from complying with requirements in the task in relation to **component testing**
- 40% of the marks (2/5) received from complying with requirements in the task in relation to **ui testing**
- Describe your approach to testing in **TESTING.md**. This section will ONLY be marked on things described in that file.

## 6. 7. Accessibility (5%)

- Your application follows standard accessibility lessons covered in lectures.
- Describe any attempts you've made to improve the Accessibility in **A11Y.md**. This section will ONLY be marked on things described in that file.

## 6. 8 Deployment (5%)

- You need to follow the instructions given in **deployment.md** to deploy your web app onto Vercel.
- When tutors mark this section, they will navigate to the **FE\_DEPLOYED\_URL** you provided in **progress.csv**, as long as your web app (frontend) is live on that url, you will gain full mark for this. :)

## 6. 9. Bonus Mark (5%)

- Implementation of extra features that are not included in the spec.

- Extra features should be non-trivial, have a clear justification for existing, and show either a form of technical, product, or creative flare. They can be functional (added behaviour) or aesthetic (making things very pretty).
- Any extra features written down in **BONUS.md** in the project folder
- Any bonus marks that extend your ass4 mark above 100% will bleed into other assignment marks, but cannot contribute outside of the 80% of the course that is allocated for assignment marks
- <b>Expectations placed on solo groups will be half of that of pairs to achieve the same mark.</b>
- If you are working individually and complete all features marked as 🎉🎉🎉 (and high quality) you can receive full marks for bonus marks.