

Foundations Capstone Project

Overview

Basic Requirements

- Planning documentation
- Needs to have an interactive front end (inputs, buttons, etc.)
- Should have at least 3 main features
- Includes styling
- Connects to a server and makes requests from the front end (express and axios)
- 2-3 minute presentation

Checklist

Overview

Remember: you do not have to include every single item! 14 is the minimum.

The **required** items are starred.

Planning

- project includes a wireframe for each view
- project includes a list of MVP features *
- project includes a data model

MVP

- app has at least 3 main features *
- front end makes a request to the server and handles the response *
- front end is interactive *
- app has custom styling *

Front End

- app has at least 5 semantic tags *
- app includes 1 view *
- app includes 1+ additional view(s)
- styling includes flexbox
- at least 1 view is responsive
- styling includes animations

Server

- app includes a GET endpoint and handler function *
- app includes a POST endpoint and handler function *
- app includes a PUT endpoint and handler function
- app includes a DELETE endpoint and handler function
- app utilizes Sequelize
- project includes at least 1 controller file

Database

- project includes a seed file or function
- app uses 1 table
- app uses 1+ additional tables
- app uses a foreign key and join

Presentation

- discusses project purpose and demonstrates MVP *
- does not discuss broken/unimplemented features
- recording is between 2-3 minutes

MVPs: add items w/date stamp, view - categories & expanded, delete, change from future/to do, currently doing, to past/done

Extras - login w/password, add & change ratings, add tags allow filtering, possibly track anything with custom categories,

To Code:

- ☐ Homepage to 'login'
- ☐ Create new user option
- ☐ Home page with recent, in progress, navigation
- ☐ View all 'wish list'
- ☐ View all 'finished list'
- ☐ Add entries form
- ☐ Set up front end with block & buttons?!
- ☐ Make buttons functional
- ☐ Set up bit.io database
- ☐ Set up back end to handle requests / send/add & view data, & delete

Tasks table - relationships user #s - categories #s