CSE186 Summer 2024

Assignment 4

In this assignment you will write a React and Material-UI Responsive Web App based on wire-frame designs given to you by a supposed client.

To encourage good practices that will serve you well in future assignments, especially the mini project, you must use React State and Context and must not use React Props more than twice.

This assignment is worth 10% of your final grade.

Installation

See instructions in Assignment 1 and ensure you are running the LTS version of Node.js.

Setup

Download the starter code archive from Canvas and expand into an empty folder. I recommend, if you have not already done so, creating a folder for the class and individual folders beneath that for each assignment.

The starter code archive contains one file you will modify:

```
        src/App.jsx
```

Do not modify any other files.

Adding JavaScript / JSX files for new components to the `src` folder is necessary, you can also create sub folders of `src` but do not add any additional tests or modify the existing tests in `src/__tests__`.

To setup the development environment, ***navigate to the folder where you extracted the starter code*** and run the following command:

```
        $ npm install
```

This will take some time to execute as `npm` downloads and installs all the packages required to build the assignment.

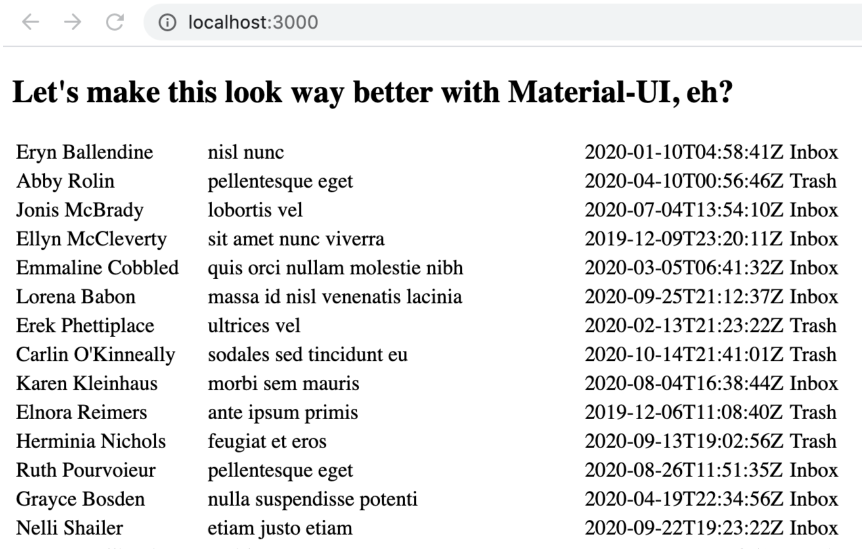
To execute all tests, run the following command:

```
        $ npm test
```

To start the dev server, run the following command:

```
$ npm run dev
```

In a new browser tab open:
`http://localhost:3000`



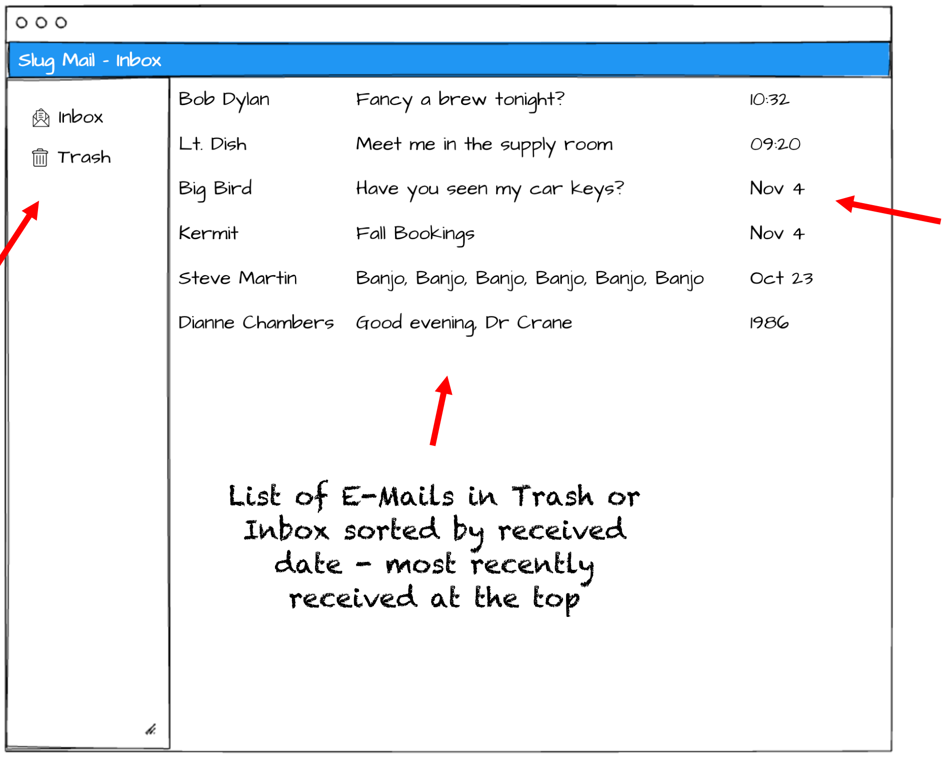
Let's make this look way better with Material-UI, eh?

Eryn Ballendine	nisl nunc	2020-01-10T04:58:41Z Inbox
Abby Rolin	pellentesque eget	2020-04-10T00:56:46Z Trash
Jonis McBrady	lobortis vel	2020-07-04T13:54:10Z Inbox
Ellyn McCleverty	sit amet nunc viverra	2019-12-09T23:20:11Z Inbox
Emmaline Cobbled	quis orci nullam molestie nibh	2020-03-05T06:41:32Z Inbox
Lorena Babon	massa id nisl venenatis lacinia	2020-09-25T21:12:37Z Inbox
Erek Phettiplace	ultrices vel	2020-02-13T21:23:22Z Trash
Carlin O'Kinneally	sodales sed tincidunt eu	2020-10-14T21:41:01Z Trash
Karen Kleinhaus	morbi sem mauris	2020-08-04T16:38:44Z Inbox
Elnora Reimers	ante ipsum primis	2019-12-06T11:08:40Z Trash
Herminia Nichols	feugiat et eros	2020-09-13T19:02:56Z Trash
Ruth Pourvoieur	pellentesque eget	2020-08-26T11:51:35Z Inbox
Grayce Bosden	nulla suspendisse potenti	2020-04-19T22:34:56Z Inbox
Nelli Shailer	etiam justo etiam	2020-09-22T19:23:22Z Inbox

Requirements

A client has given you a set of wireframes / mock-ups for an e-mail reader they want you implement in React and Material-UI. Your client has several requirements…

Desktop 1:



Bob Dylan	Fancy a brew tonight?	10:32
Lt. Dish	Meet me in the supply room	09:20
Big Bird	Have you seen my car keys?	Nov 4
Kermit	Fall Bookings	Nov 4
Steve Martin	Banjo, Banjo, Banjo, Banjo, Banjo, Banjo	Oct 23
Dianne Chambers	Good evening, Dr Crane	1986

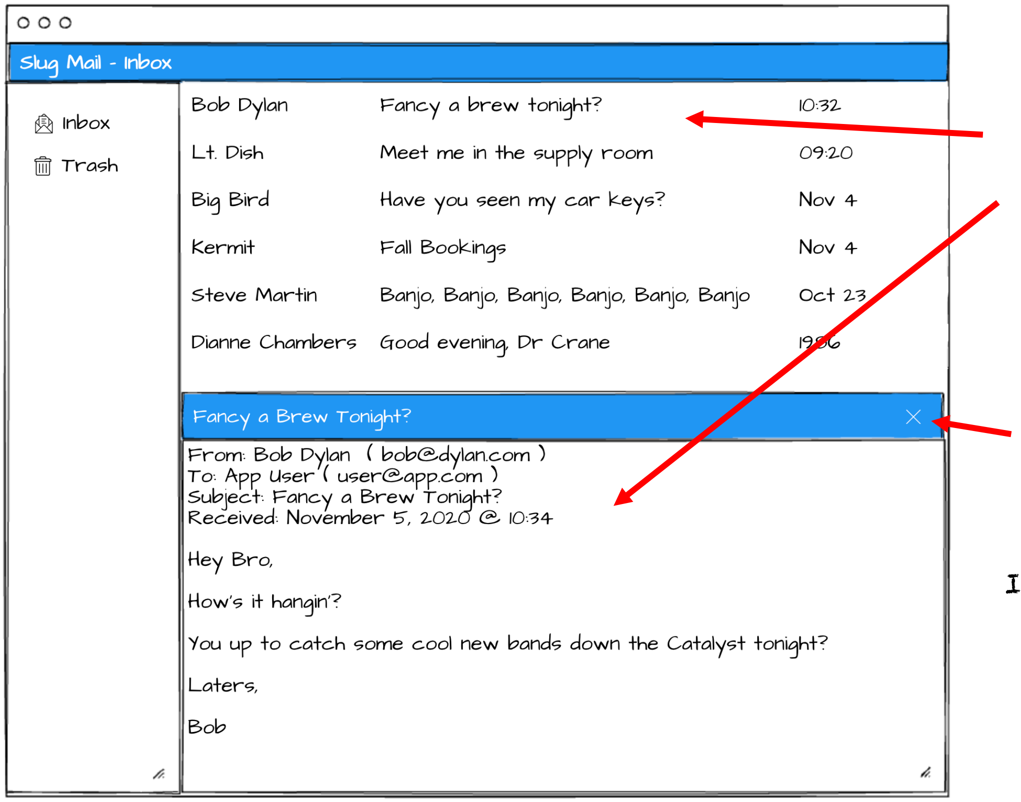
Title: Shows if we're in Trash or Inbox

Button Bar: Click items to switch between Trash and Inbox in the E-mail list

List of E-Mails in Trash or Inbox sorted by received date - most recently received at the top

Show time if received today, Month and Day if in last twelve months, Year only if more than twelve months ago - thanks!

Desktop 2:



Bob Dylan	Fancy a brew tonight?	10:32
Lt. Dish	Meet me in the supply room	09:20
Big Bird	Have you seen my car keys?	Nov 4
Kermit	Fall Bookings	Nov 4
Steve Martin	Banjo, Banjo, Banjo, Banjo, Banjo, Banjo	Oct 23
Dianne Chambers	Good evening, Dr Crane	1986

Fancy a Brew Tonight?

From: Bob Dylan (bob@dylan.com)
To: App User (user@app.com)
Subject: Fancy a Brew Tonight?
Received: November 5, 2020 @ 10:34

Hey Bro,

How's it hangin'?

You up to catch some cool new bands down the Catalyst tonight?

Laters,

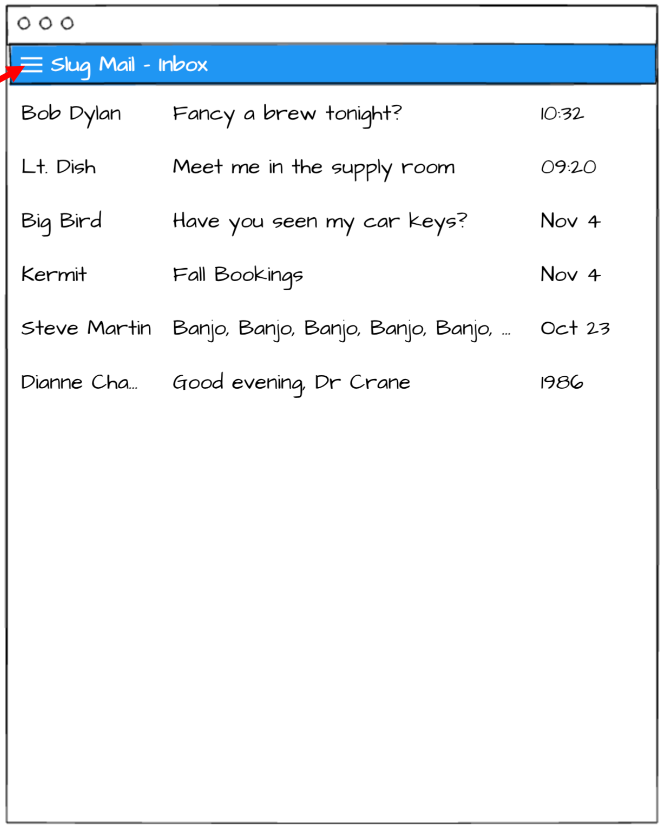
Bob

Click an E-Mail in the list and the full message appears in a viewing box below the list

Click this 'X' (or hit 'Esc') and the viewing box goes away - Also hide the viewing box if the user switches from Inbox to Trash and visa-versa

Desktop 3:

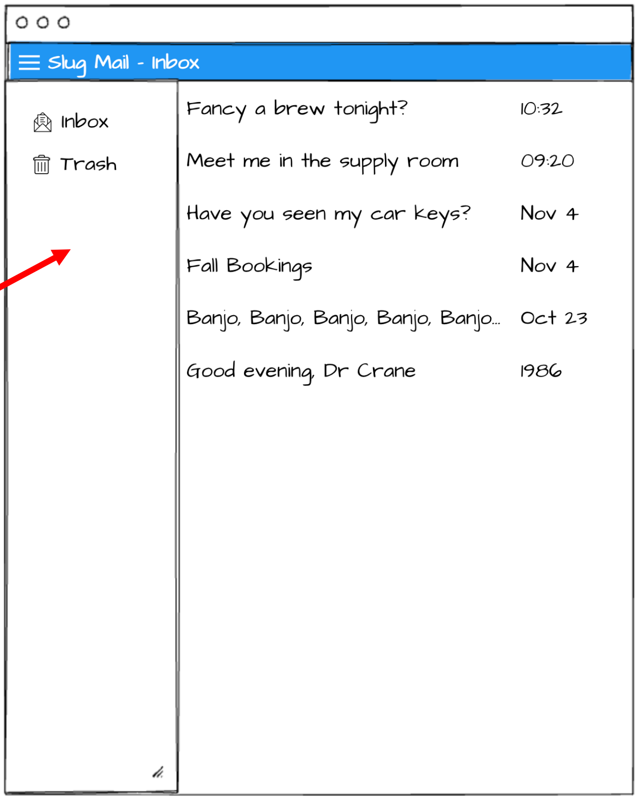
When the window is reduced in width to the point where the message titles are getting too short, hide the Button Bar and replace it with a Menu Icon to the left of the title



☰ Slug Mail - Inbox

Bob Dylan	Fancy a brew tonight?	10:32
Lt. Dish	Meet me in the supply room	09:20
Big Bird	Have you seen my car keys?	Nov 4
Kermit	Fall Bookings	Nov 4
Steve Martin	Banjo, Banjo, Banjo, Banjo, Banjo, ...	Oct 23
Dianne Cha...	Good evening, Dr Crane	1986

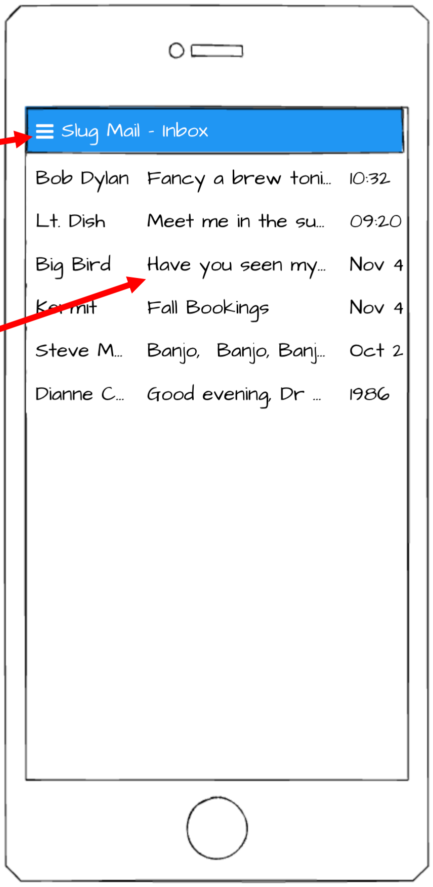
If the menu Icon is clicked, show the Button Bar over the top of the E-Mail List. Clicking on the list or Title or Menu Icon should hide the Button Bar again. Clicking a Button Bar item should select the appropriate list AND close the button bar ☺



☰ Slug Mail - Inbox

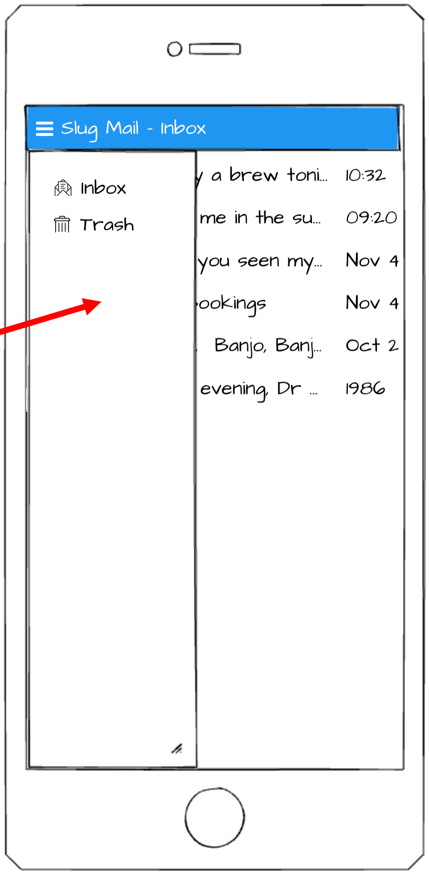
Fancy a brew tonight?	10:32
Meet me in the supply room	09:20
Have you seen my car keys?	Nov 4
Fall Bookings	Nov 4
Banjo, Banjo, Banjo, Banjo, Banjo...	Oct 23
Good evening, Dr Crane	1986

Mobile 1:


Slug Mail - Inbox
Bob Dylan Fancy a brew toni... 10:32
Lt. Dish Meet me in the su... 09:20
Big Bird Have you seen my... Nov 4
Kermit Fall Bookings Nov 4
Steve M... Banjo, Banjo, Banj... Oct 2
Dianne C... Good evening, Dr ... 1986

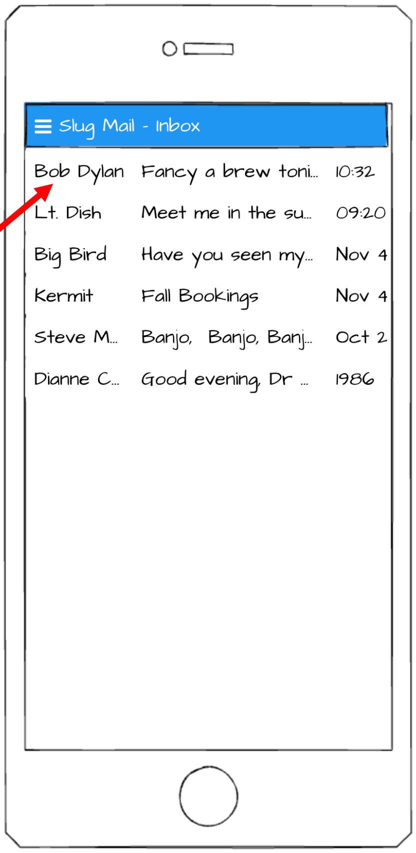
Show a menu icon next to the Title

Show the Trash or Inbox E-Mails using the full width of the phone screen



If the menu Icon is clicked, show the Button Bar over the top of the E-Mail List. Hitting 'Esc' and clicking on the list or Title or Menu Icon should hide the Button Bar again. Clicking a Button Bar item should select the appropriate list AND close the button bar - woo-hoo!

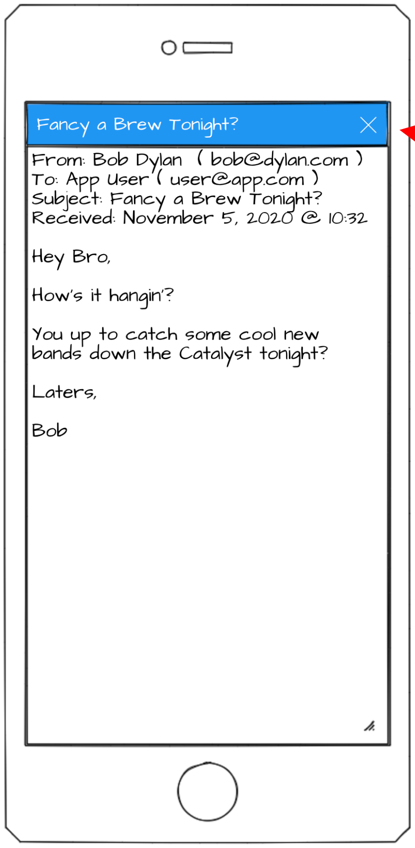
Mobile 2:



☰ Slug Mail - Inbox

Bob Dylan	Fancy a brew toni...	10:32
Lt. Dish	Meet me in the su...	09:20
Big Bird	Have you seen my...	Nov 4
Kermit	Fall Bookings	Nov 4
Steve M...	Banjo, Banjo, Banj...	Oct 2
Dianne C...	Good evening, Dr ...	1986

Click an E-Mail in the list and the full message appears and takes over the whole screen with the subject in the title bar



Fancy a Brew Tonight? ✕

From: Bob Dylan (bob@dylan.com)
To: App User (user@app.com)
Subject: Fancy a Brew Tonight?
Received: November 5, 2020 @ 10:32

Hey Bro,

How's it hangin'?

You up to catch some cool new bands down the Catalyst tonight?

Laters,

Bob

Click this 'X' (or hit 'Esc') and the E-Mail Viewer goes away and the E-Mail List is shown again

What steps should I take to tackle this?

Study the examples at https://mui.com/material-ui/all-components/ and play with their code in the provided sandboxes, then use some of it as starter code for your Web App. Don't be alarmed if you end up throwing a few attempts away, that's perfectly normal and a useful experience.

You'll probably find you don't understand all the code you use, but that's ok. Get it running then modify so it does what you want. Don't struggle for too long with something you can't get to work. Find another example and try that.

Some guidelines:

- Write functional React components with Hooks, not ECMAScript Classes.
- Start simple and work up from there, a reasonable initial approach might be:
  - Create a simple email-list component
    - Initially show all emails
    - Ensure list renders well on desktop and mobile
  - Introduce state for which mailbox to view
    - Set the state in code
    - Check list is filtered correctly to only show Inbox or Trash
  - Add a permanent drawer with buttons to select which mailbox to view
    - You'll need a `Context` to share state between components

Lint-as-you-type

You'll notice that the assignment has been configured so your React Web App will not run if your code fails any of the linter checks. Your code will therefore need to be super clean and tidy before you can manually test it ☺

Much of the code you find on-line will fall foul of the linter checks - so fix it!

Testing

The suplied tests use React Testing Library https://testing-library.com/docs/ and the matchers provided by Jest DOM https://github.com/testing-library/jest-dom#usage. You do not need to write any of your own tests, but you may find it useful to study the supplied tests as you will need to write your own for your mini project.

To run the supplied tests continually as you develop:

```
$ npm run watch
```

To execute all tests once, run the following command:

```
$ npm test
```

To run the linter outside of the React development server:

```
$ npm run lint
```

How much code will I need to write?

Potentially very little. Use the examples at https://mui.com/material-ui/all-components/ and play with their code in the provided sandboxes, then use some of it as starter code for your Web App. And anywhere else you can find code that helps you meet the requirements. Remember to give credit to the original authors.

Grading scheme

The following aspects will be assessed:

1. (50%) **Does it work?**

   a. Desktop Tests (30%)
   b. Mobile Tests (20%)

2. (50%) **Is it well written?**

   a. No linter errors or warnings (0%) tests will not run if you have any
   b. Code Coverage (statements, lines, branches, functions) (50%)

100%	(50%)
>= 98%	(30%)
>= 95%	(10%)
< 95%	(0%)

3. (-100%) **Is it *badly* written?**

   Any of the following errors incurs the entire deduction:

   a. Not using Material UI Components
   b. Using Props more than twice
   c. Not using State and Context
   d. Suppressing eslint errors/warnings

4. (-100%) **Did you give credit where credit is due?**

   a. Your submission is found to contain code segments copied from on-line resources and you failed to give clear and unambiguous credit to the original author(s) in your source code (-100%). You will also be subject to the university academic misconduct procedure as stated in the class academic integrity policy.

   b. Your submission is determined to be a copy of a past or present student's submission (-100%)

   ~~c. Your submission is found to contain code segments copied from on-line resources that you did give a clear an unambiguous credit to in your source code, but the copied code constitutes too significant a percentage of your submission:~~

~~o < 25% copied code~~	~~No deduction~~
~~o 25% to 50% copied code~~	~~(-50%)~~
~~o > 50% copied code~~	~~(-100%)~~

In this assignment you are encouraged to use as much code as you can find on-line - but remember to give credit to the original authors.

What to submit

Run the following command to create the submission archive:

```
$ npm run zip
```

**** UPLOAD** `CSE186.Assignment4.Submission.zip` **TO THE CANVAS ASSIGNMENT AND SUBMIT ****