# *WOW! Case Study Project*

Image source: https://pixabay.com/photos/zumba-party-marathon-sport-4333580/

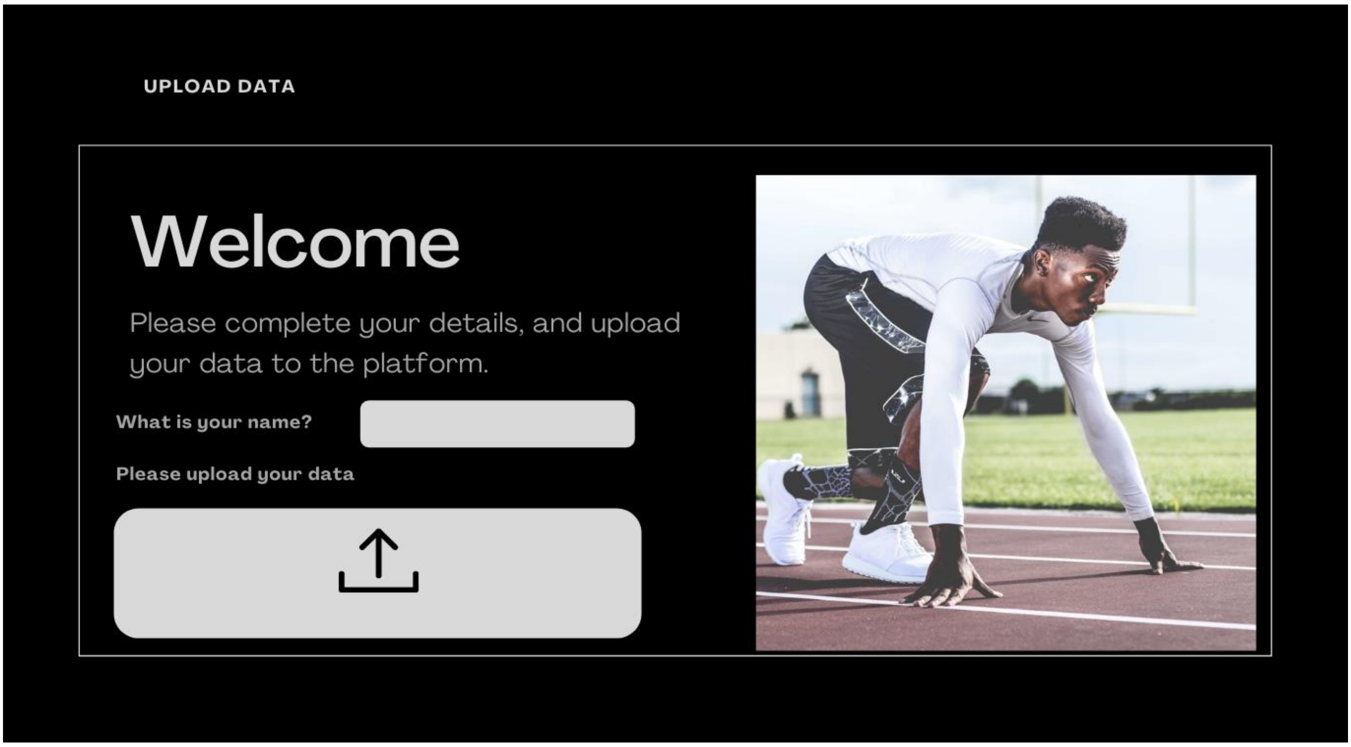
# Name: *Vusani Makhomu*

# Table of contents

## Project Description

This was a case study project to test my coding abilities. I was given 3 PNG wireframes that I had to build. Following are the wireframes that I was given:

### Wireframe 1:



UPLOAD DATA

Welcome

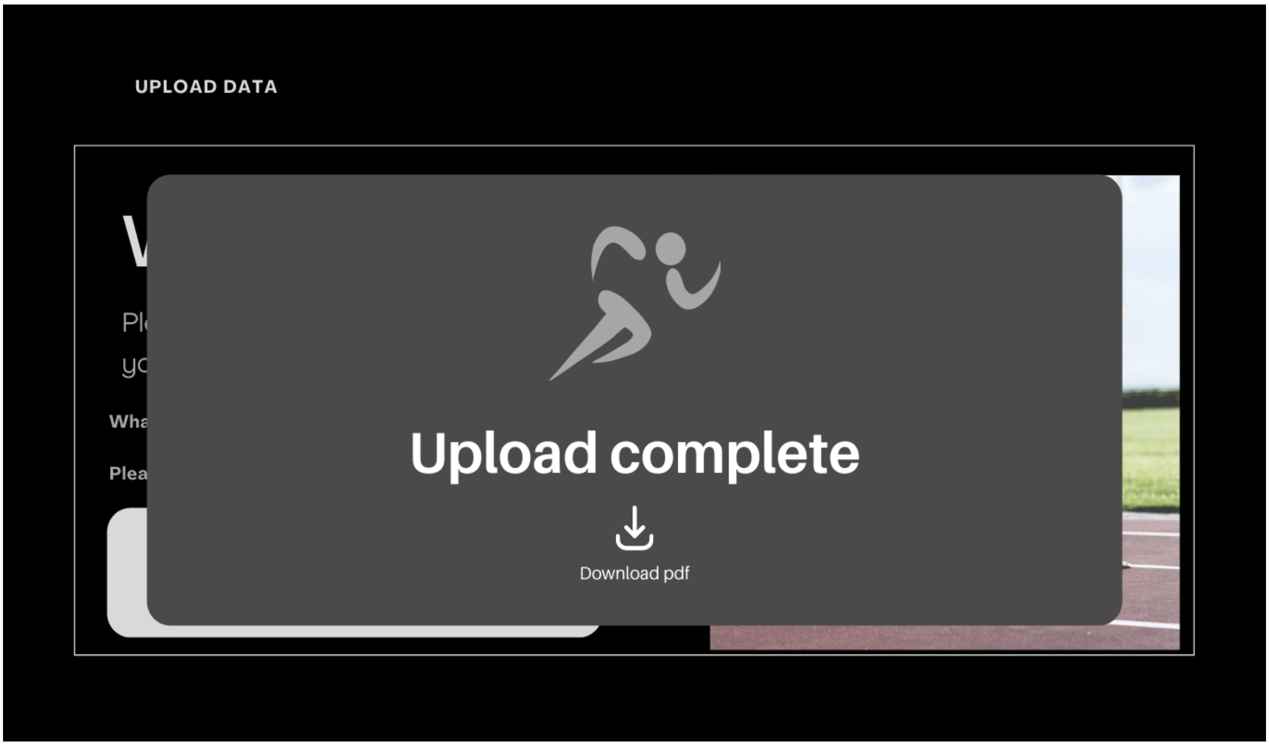
Please complete your details, and upload your data to the platform.

What is your name?

Please upload your data

The above figure is to be the landing page.

### Wireframe 2:



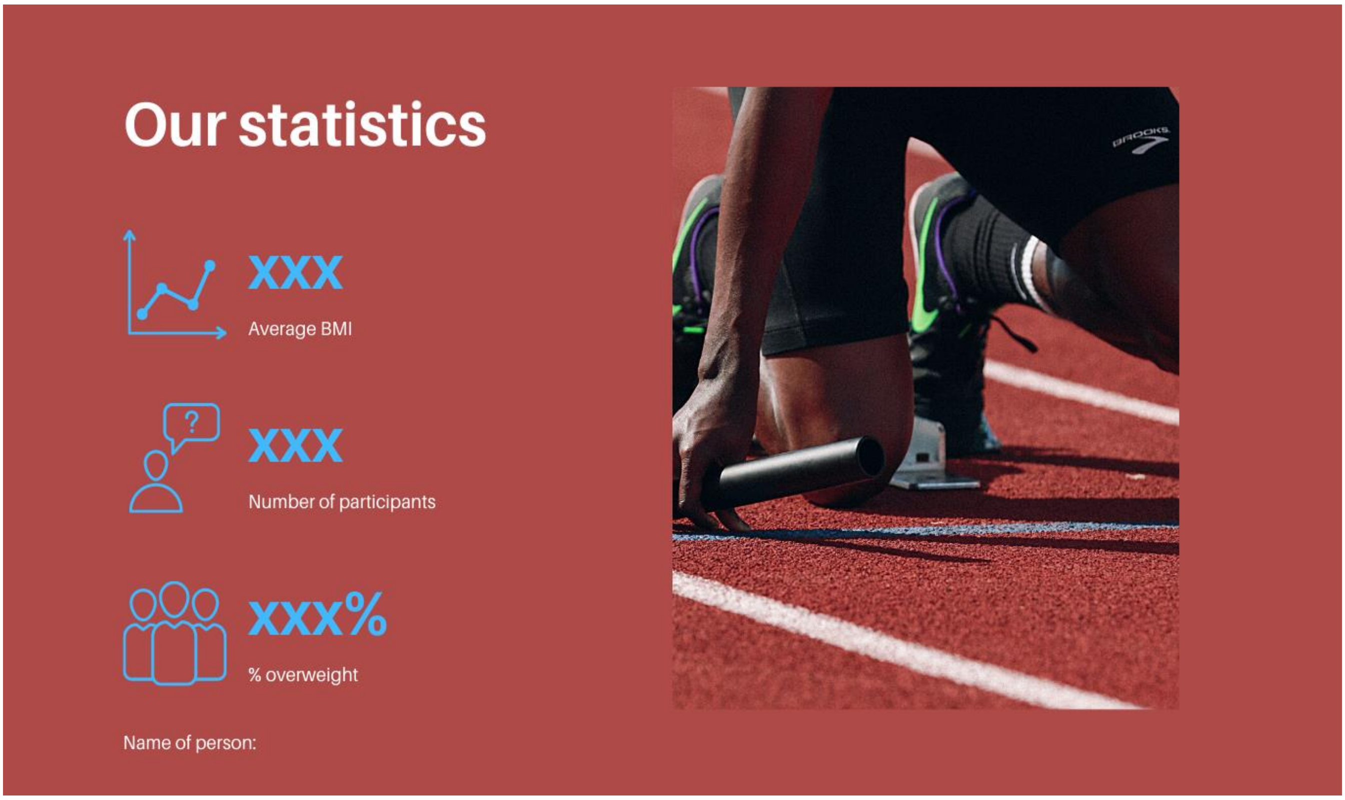
UPLOAD DATA

Upload complete

Download pdf

The above figure is to be the pop up that alert the user that their upload has completed. In addition, they should be provided with a button that allows them to download a pdf of their results.

## Wireframe 3:



Our statistics

XXX

Average BMI

XXX

Number of participants

XXX%

% overweight

Name of person:

The above figure is to be the contents of their pdf. The pdf should display the average BMI (calculated as weight/height^2, the number of participants, the percentage of participants who are overweight (who have a BMI > 25.0).

In addition to the above wireframes, I was given a dataset that I was to use during development. I was supposed to be given the necessary icons and images for the project, however this never happened due to technical issues. I had to improvise.

## Project implementation

I used React to build this project. The code for the project is all located in *src/index.js*. The stylings are in *src/index.css*. There are 2 components:

- Form (child component)
- Landing Page (parent component)

### Parent component

The parent component is responsible for letting the child component know what to render. It does this by passing it a variable called showPopUp. This variable contains a boolean. If its value is false, then the child component will render the form that allows the user to enter their name and upload their file. If its value is true, the child component will render the pop-up that allows the user to download a pdf.

### Child component

Since the file upload, user entering their name and pdf download happens inside the child component, the child component first checks if a user has uploaded their file and entered their name before allowing them to download a pdf. If they haven't, the component will display an alert telling them to go back and enter their name and upload their file. When a file upload happens, a OnChange event is captured. Note that the expected file type is .csv.

The event handler for the OnChange event processes the file. First, it extracts the headers and rows of the file. Secondly, the data inside the rows is processed and stored in an array as a dictionary. Each row becomes a dictionary object. The header (column name) is the key, and the row value is the value. For example, the provided csv contains the following columns: Gender, Height, Weight, Index. The first row contains the following data: Male, 174, 96, 4. During processing, the resulting dictionary will be: {Gender: Male, Height: 174, Weight: 96, Index: 4} This object will then be added into an array.

From this array data, the sum BMI will be calculated. From this, the average BMI will be calculated. The number of participants is the length of the array. All this information will be stored in the state of this component. Lastly, this information will be used to generate the resulting pdf.

## Problems encountered

Initially, the plan was to use react-pdf/renderer (https://react-pdf.org/) to render the pdf. This library is quite flexible and can allow beautiful pdfs to be created. In addition, it also allows styling of the pdf content. The styling can be done using css. However, the library has no support for the current version of react (18).

To work around this problem, I decided to downgrade from react (18) to react (17). Then I found out that the library required the version of webpack to be < 5. To work around this problem, I decided that I'll also downgrade the version of webpack. Downgrading the version of webpack proved to be difficult. In addition, it seemed that this was not going to solve the problem. The library (pdf renderer) was dependent on npm to install polyfills by default. This was no longer the case.

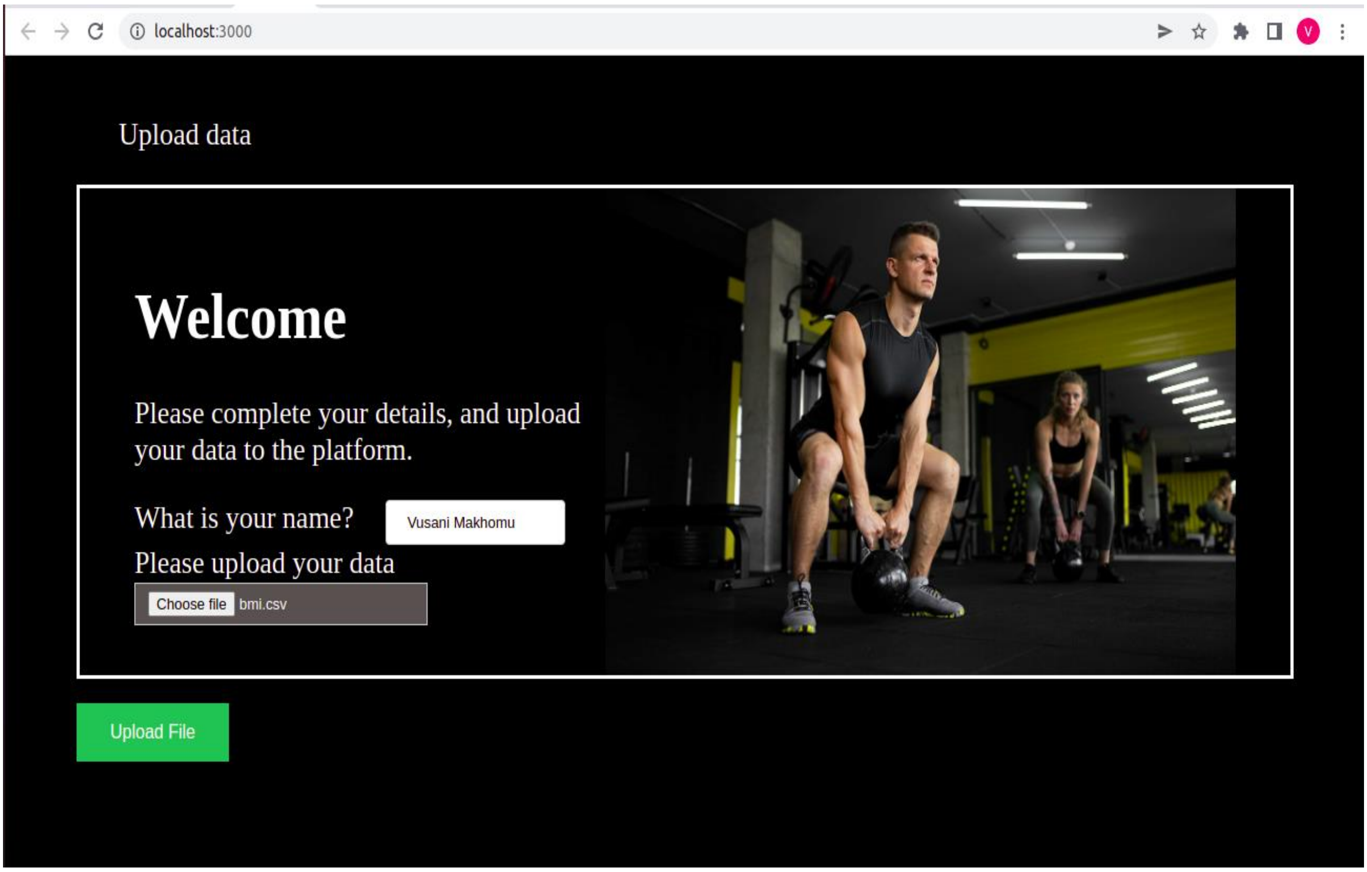
In addition to downgrading webpack, it looked as though I would need to install polyfills. Then I found out I also needed to install buffer, stream and 3 other packages. All this was taking time away from the project. In the end, I decided to drop this library. It meant that I would need to find another library for creating pdf. I found jsPDF. Adding images to this pdf is a hassle as you first need to convert the image to a Data URL before you can add it to the pdf. The website referenced for doing this seems to no longer be working.

This is the reason the final pdf contains no images. There is no documentation on the jsPDF website on how you can color a page. This is the reason the resulting pdf page has no color. However, I figured that it was the data that was important, so I decided to only add that to the resulting pdf.

## Final Project

Following are the screenshots of the final project.

### Landing page:



Upload data

Welcome

Please complete your details, and upload your data to the platform.

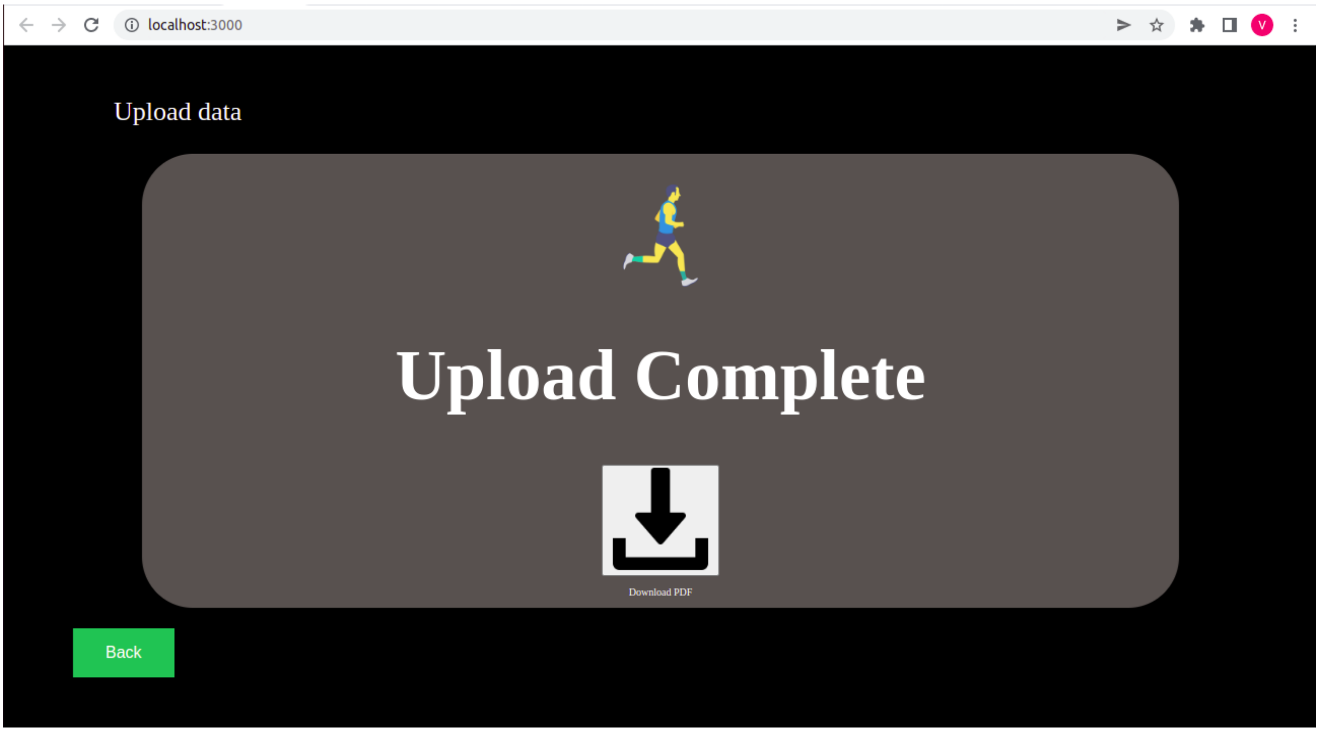
What is your name? Vusani Makhomu

Please upload your data

Choose file bmi.csv

The above figure is the landing page. I added an “Upload File” button that a user can click on after they have uploaded their file.

### Upload Pop Up:

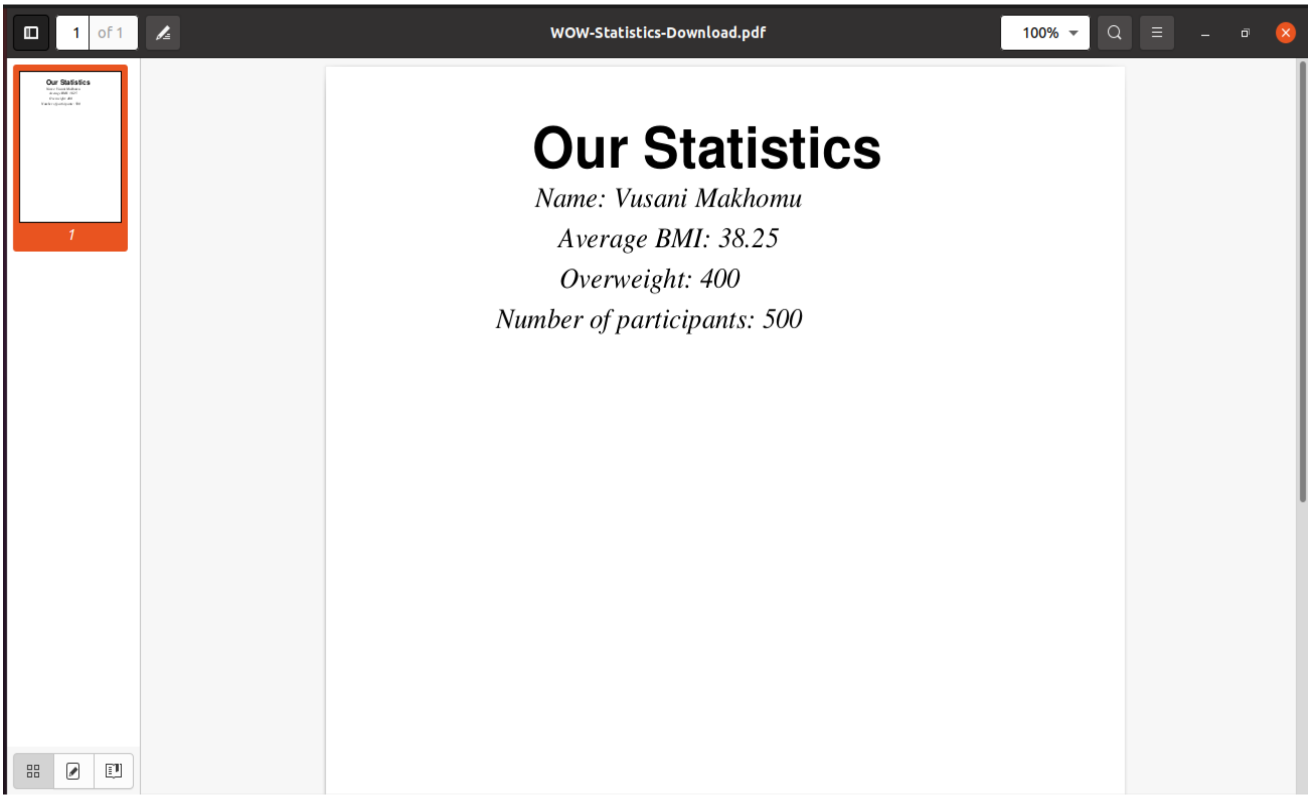


Upload data

Upload Complete

The above figure is the pop-up page. This is shown after a user has clicked on the “Upload File” button. I also added a “Back” button that the user can click on to go back to the Landing Page. Clicking on the download icon downloads the PDF.

### Overview of PDF



# Our Statistics

*Name: Vusani Makhomu*

*Average BMI: 38.25*

*Overweight: 400*

*Number of participants: 500*

The above figure is the contents of the downloaded PDF.

## GitHub Repository Link:

https://github.com/Vusani-Makhomu/WOW-Project

## References:

- https://react-pdf.org/ (The React-pdf official website)
- https://github.com/parallax/jsPDF (The Github repository for jsPDF)
- https://parall.ax/products/jspdf (The official website for jsPDF)
- https://rawgit.com/MrRio/jsPDF/master/docs/index.html (The official documentation for jsPDF)

- https://stackoverflow.com/questions/48395804/where-is-create-react-app-webpack-config-and-files (Stackoverflow post for finding webpack files)
- https://stackoverflow.com/questions/41661383/how-to-determine-the-installed-webpack-version (Stackoverflow post for finding the installed version of webpack)
- https://stackoverflow.com/questions/65922760/react-build-not-found-error-cant-resolve-buffer (Stackoverflow post for resolving buffer issues with the react-pdf)
- https://stackoverflow.com/questions/70729090/i-cant-modify-webpack-config-in-create-react-app-to-install-react-pdf (Stackoverflow post for modifying the webpack.config.js file)
- https://stackoverflow.com/questions/43589964/how-to-downgrade-version-of-webpack (Stackoverflow post for dowgrading webpack)
- https://github.com/diegomura/react-pdf#webpack-5 (Possible solution for working around the webpack issue for react-pdf)
- https://github.com/diegomura/react-pdf/issues/1645 (Possible solution for installing the stream package for react-pdf)