

IMY 220

Assignment 3

React Basics

General Instructions

- This assignment must be completed and submitted by the due date which is available on ClickUP.
- This assignment is a take-home assignment and may take one or two days to complete.
- This assignment should be completed on your own, but you may come to the tutorial session or email the assistant lecturer if you need further assistance.
- **No late / email submissions will be accepted.**

This assignment focuses on **React Basics (Creating and Using Components, Component State)**. You will be required to create a simple React application that displays a feed of events and allows a user to search for an event.

Download the provided files from ClickUP. You have been provided with an HTML file (**index.html**) that has code for using React with a Babel loader in the browser. **Note:** *this approach is not generally recommended for production projects as it is very slow. This is just to help you learn React fundamentals before using webpack.*

You have also been provided with a JavaScript file (**App.js**) that has an array of events as JavaScript objects. For this assignment, you will need to create a feed that showcases these events in reverse chronological order, as well as a search bar for searching for different events. Follow the instructions below to create all the components and functionality you need for this assignment.

Note: *You would usually create each of your components in separate files, and export/import them to use them. For simplicity's sake, and because Babel browser loader doesn't support multiple files, implement all these components in **App.js**.*

Part 1 - Event Component

Create a class component called **Event** that displays a single event as follows:

A Walk in the Park

2021-09-19

Let's go walking and feed the ducks. #ducks #walk #park #Sunday

The component must be **dynamic**, i.e., it should take in values as props and display these.

Render this component in the “#root” div in the browser to test your code is working before moving on to the next step.

Part 2 – EventFeed Component

Create a class component called **EventFeed** that displays a list of events in **reverse chronological order** (i.e., most recent date first) as follows:

Comicon Anyone?

2024-09-26

Who's going to #comicon this year? Let's #meetup - I'll be Spiderman!

Crochet Date!

2024-07-09

Let's meetup to go crochet in the park. I'll bring the wool!! #park #crochet #meetup

Farmers Market

2024-04-01

Fresh produce, homemade goods, and more at the local farmers market this week #farmersmarket #organic

Live Concert: Rock the Night

2023-06-25

Enjoy an electrifying night of live music from your favourite #rock artists #concert #rockmusic #livemusic

Star Wars Under the Stars

2023-05-04

Watch your favorite #StarWars movies under the open sky! #movienight #outdoor #maythe4thbewithyou

Hackathon

2023-03-10

Compete in this year's #hackathon to win amazing prizes and meet fellow #coders #coding

The component should take an array of event objects as a prop and render out these events using the **Event** component from Step 1.

Render this component in the “#root” div in the browser to test your code is working before moving on to the next step.

Part 3 – Search Component

Create a class component called **Search** that looks as follows:

Search

When the search button is clicked, OR when enter is pressed (it should support both methods – *hint: <form> has an event for this*), the **search() function** (defined in **this** component) should be called. This function should make a call to **handleSearch()** that is **not** defined in this component but instead passed in as a prop. It should call **handleSearch() with the search term from its input**.

Part 4 – App Component

*It is convention to create React Apps that have one root component at the highest level that holds all the other components. It doesn't usually have any other functionality besides passing data between its children. Typically, this component is called **App**, which is the convention we will follow.*

Create this component and **cut and paste the provided 'events' array inside**. This will act as our 'backend' since we are not making any API calls in this assignment (*typically you would be getting this data from some kind of backend, but we will ignore that for now*).

This component also has a **state** that keeps track of the state of the feed before/after searching. **Initialise** this with the events array you cut and pasted in (this state will be passed to your **EventFeed** component).

Also define a function inside the **App** component called **searchFeed()** with the following functionality (this will be passed to your **Search** component):

Note: Do not modify the events array for this, as this will not work. You must make use of state for this next section.

1. If the search term is **empty**, the feed should be **reset** to what is contained in the events array.

2. If the search term **starts with a #**, the feed should show all the events from the events array whose **descriptions** contain a matching #tag as the search term.
3. **Otherwise**, the feed should return all the events whose **titles** contain the search term.

This component should hold the **EventFeed** and **Search** components you created in the previous steps and should pass each child the required props they need.

Render this component in the “#root” div in the browser. This component should look as follows initially:

Events!

Search

Search

Feed

Comicon Anyone?

2024-09-26

Who's going to #comicon this year? Let's #meetup - I'll be Spiderman!

Crochet Date!

2024-07-09

Let's meetup to go crochet in the park. I'll bring the wool!! #park #crochet #meetup

Farmers Market

2024-04-01

Fresh produce, homemade goods, and more at the local farmers market this week #farmersmarket #organic

Live Concert: Rock the Night

2023-06-25

Enjoy an electrifying night of live music from your favourite #rock artists #concert #rockmusic #livemusic

Star Wars Under the Stars

2023-05-04

Watch your favorite #StarWars movies under the open sky! #movienight #outdoor #maythe4thbewithyou

When a user searches for **#summertime**, the component should look like this:

Events!

Search

Search

Feed

Summer Braai

2021-08-05

Come and enjoy a delicious braai with friends and family #braai #summertime #summer #fun

Beach Day!

2019-12-28

Let's have a fun day on the beach right before #xmas !! #beachday #summertime

If the user *then* searches for “**Park**” (with a capital P), the component should look like this:

Events!

Search

Search

Feed

A Walk in the Park

2021-09-19

Let's go walking and feed the ducks. #ducks #walk #park #Sunday

Finally, if the user **backspaces** their search until it is **blank** and searches (either by pressing enter or clicking the search button), the component should look like it did initially.

Keep in mind the page should **not** reload when the user searches for a term.

Submission Instructions

Place these in a folder named A3_u12345678 where 12345678 is your student number.

- index.html
- App.js

Zip the **folder** and upload this to ClickUP in the relevant submission slot before the deadline.