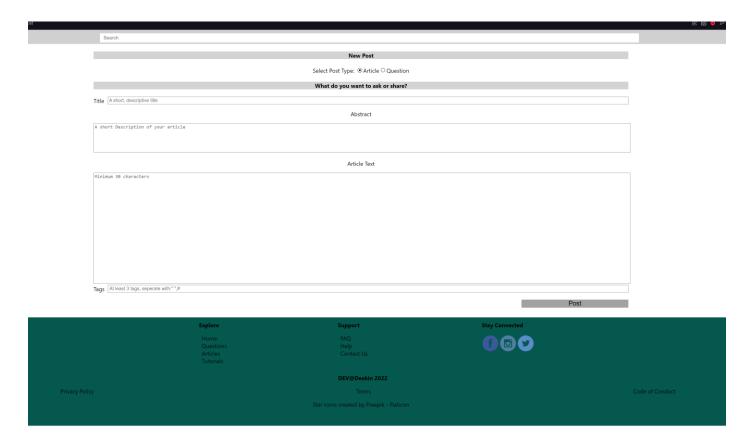
Video: https://video.deakin.edu.au/media/t/1 1lf4xvse

Git: https://github.com/CameronHart1/SIT313-FullStack

Screenshots:



```
handleTextChange={(val) => handleContentChange(val, 1)}
title={"Abstract"}
      <LargeTextBox</pre>
   // Conditional rendering for abstract Box const abstractBox = article ? (
      link.href = jsonString;
link.download = `${article ? "Article" : "Question"}_Post.json`;
     ))';
const link = document.createElement("a");
itek bnef = isonString;
         JSON.stringify(jsonOBJ)
      // https://codesandbox.io/s/4t2xb?file=/src/App.js:1112-1382 const jansfring = 'data:text/json;chatset=utf-8,${encodeURIComponent(
            content: content[2],
tags: content[3],
1.
              title: content[0],
abstract: content[1],
content: content[2],
tags: content[3],
      // turning data into JSON var jsonOBJ = article
   const submitPost = () => {
   // just downloading content as JSON
   const handleTagChange = (newVal) =>
  setContent([6], content[1], content[2], newVal]);
           ? [content[0], newVal, content[2], content[3]]
: [content[0], content[1], newVal, content[3]]
     setContent(
index === 1
   const handleContentChange = (newVal, index) =>
   // For storing abstract or conte
   const handleTitleChange = (newVal) =>
  setContent([newVal, content[1], content[2], content[3]]);
const MakePost = () => {
  const [content, setContent] = useState(["", "", "", []]);
  const [article, setArticle] = useState(true);
import React, { useState } from "react";
import ".../CSS/p_make_post.css";
```

```
curn (
div className="contentDiv";
  <div className="greyBox";
<h4>New Post</h4>
                                                                                                                                                                                placeholder="A short, descriptive title"
onChange={(e) => props.handleTextChange(e.target.value)}
        id="article"
name="post_type"
value="Article"
onClick={isArticle}
                                                                                                                                                                             //
const LargeTextBox = (props) => {
  const WordLimit = props.WordLimit;
  const Rows = props.Rows;
  const Col = props.Columns;
  const title = props.title;
  const placeHolder = props.placeHolder;
  const startText = props.text ? props.text : "";
         defaultChecked
        type="radio"
id="question"
name="post_type"
value="Question"
onClick={isArticle}
     {title}
{textarea
   type="text"
   id="TextInput"
   //
div>
div className="greyBox">
(h4>What do you want to ask or share?</h4>
                                                                                                                                                                                            rows={Rows}
cols={Col}
                                                                                                                                                                                            placeholder=(placeHolder}
onChange={(e) => props.handleTextChange(e.target.value)}
defaultValue={startText}
     {/* Abstract, with conditional rendering*/}
{abstractBox}
        handleTextChange={(val) => handleContentChange(val, 2)} title={article ? "Article Text" : "Describe your problem"} placeHolder={"Winimum 30 characters"} word.imit="500"
                                                                                                                                                                             const TagComp = (props) => {
                                                                                                                                                                                type="text"
placeholder='At least 3 tags, seperate with:" ",#'
// using regex to split hashtags and filtertring out any empty values
onChange={(e) =>
props.handleTextChange{
    e.target.value.split(/(?:,| |#)+/).filter((s) => s)
```

```
const PostButton = (props) => {
  const [error, setError] = useState([]);
 const errorCheck = () => {
   // checking they are above lengths / are needed
if (content[0].length < 5)</pre>
      errors.push({
       error: "NO_TITLE",
msg: "Title is too short. Minimum 5 characters",
    if (props.article && content[1].length < 20)</pre>
      errors.push({
       error: "NO_ABSTR", msg: "Abstract is too short. Minimum 20 characters",
    if (content[2].length < 30)</pre>
     errors.push({
   error: "NO_CONTENT",
   msg: "Content is too short. Minimum 30 charcaters",
    if (content[3].length < 3)
     errors.push({ error: "NO_TAGS", msg: "Not enough tags. Minimum 3 tags" });
    if (errors.length > 0) setError(errors);
     postFunction();
      setError([]);
   error.length > 0
     ? error.map((errorText) => {errorText.msg})
   <div className="SubmitPostButton">
      <button onClick={errorCheck}>Post</button>
export default MakePost;
```