

Video : [https://video.deakin.edu.au/media/t/1\\_1lf4xvse](https://video.deakin.edu.au/media/t/1_1lf4xvse)

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

Screenshots:

The screenshot shows a web application interface for creating a new post. At the top, there is a search bar. Below it, a 'New Post' button is centered. Underneath, a 'Select Post Type' section has radio buttons for 'Article' (selected) and 'Question'. A heading 'What do you want to ask or share?' is followed by a 'Title' input field with placeholder text 'A short, descriptive title'. Below the title is an 'Abstract' section with a text area containing the placeholder 'A short Description of your article'. This is followed by a large 'Article Text' text area with a 'Minimum 30 characters' warning. At the bottom, there is a 'Tags' input field with a placeholder 'At least 3 tags, separate with ""', and a 'Post' button.

```
// Conditional rendering for abstract Box
const abstractBox = article ? (
  <LargeTextBox
    handleChange={(val) => handleContentChange(val, 1)}
    title={"Abstract"}
    placeholder={"A short Description of your article"}
    wordLimit="250"
    Rows="5"
    text={content[1]}
    Columns="50"
  />
) : null;
```

[illegible]



```

},
// -----
const PostButton = (props) => {
  const [error, setError] = useState([]);
  const content = props.content;
  const postFunction = props.postFunction;

  const errorCheck = () => {
    var errors = [];
    // checking they are above lengths / are needed
    if (content[0].length < 5)
      errors.push({
        error: "NO_TITLE",
        msg: "Title is too short. Minimum 5 characters",
      });
    if (props.article && content[1].length < 20)
      errors.push({
        error: "NO_ABSTR",
        msg: "Abstract is too short. Minimum 20 characters",
      });
    if (content[2].length < 30)
      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);
    else {
      postFunction();
      setError([]);
    }
  };

  const errorText =
    error.length > 0
      ? error.map((errorText) => <p key={errorText.error}>{errorText.msg}</p>)
      : null;

  return (
    <div className="SubmitPostButton">
      <button onClick={errorCheck}>Post</button>
      {errorText}
    </div>
  );
};
// -----
export default MakePost;

```