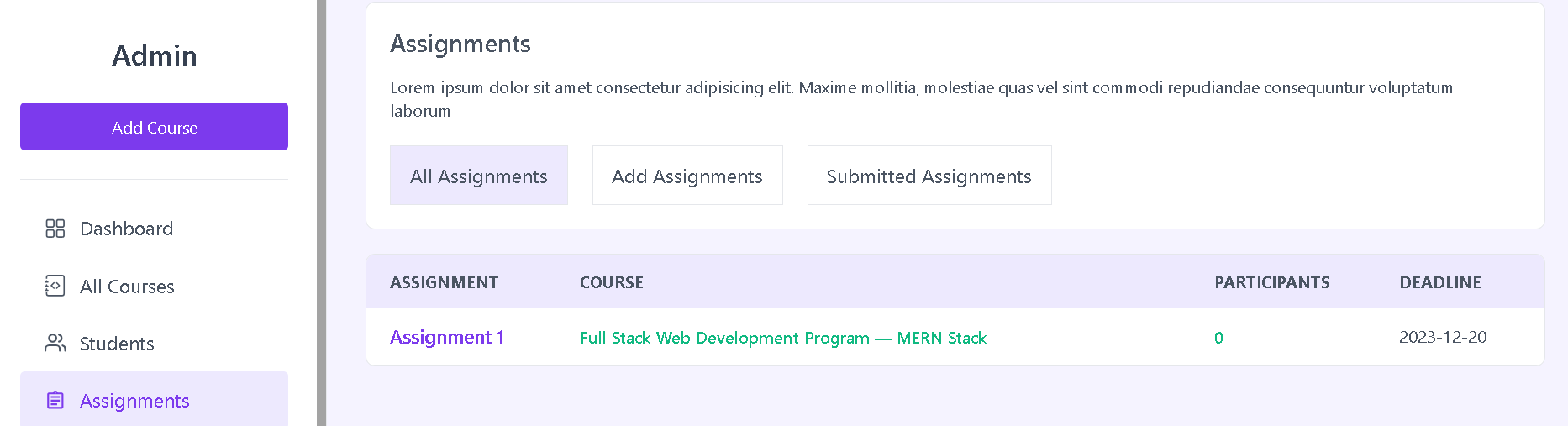
**Assignments** [**https://citria.org/profile/assignments**](https://citria.org/profile/assignments)



This is a React component that allows managing assignments. Here are the key things it does:

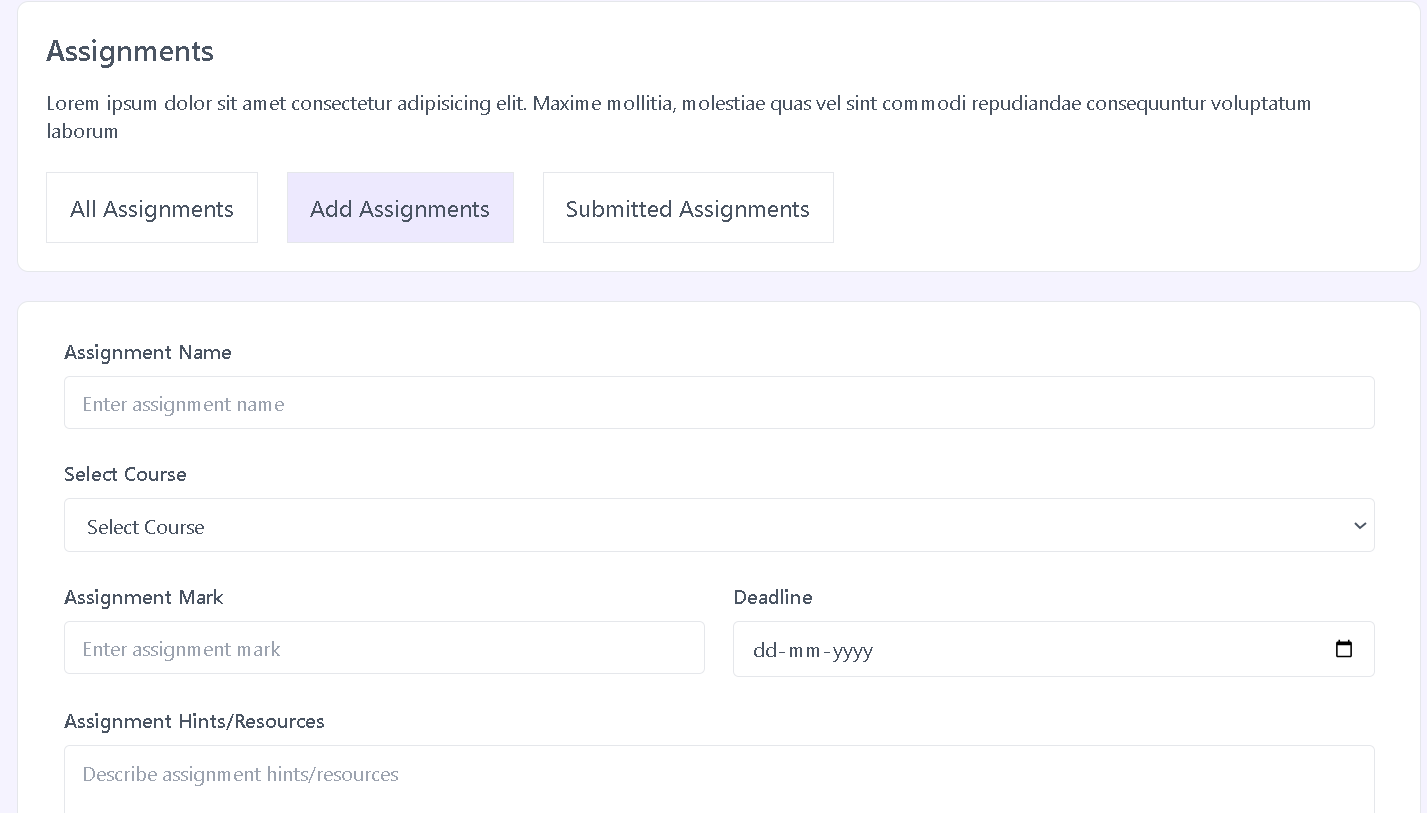
1. Defines state using the useState hook to track which view is active - "show" to show all assignments, "add" to add a new assignment, or "submit" to show submitted assignments.
2. Renders a title and intro text about assignments.
3. Shows a button group to switch between the different views. The buttons update the state to change which view is shown.
4. Conditionally renders one of three components depending on the state:

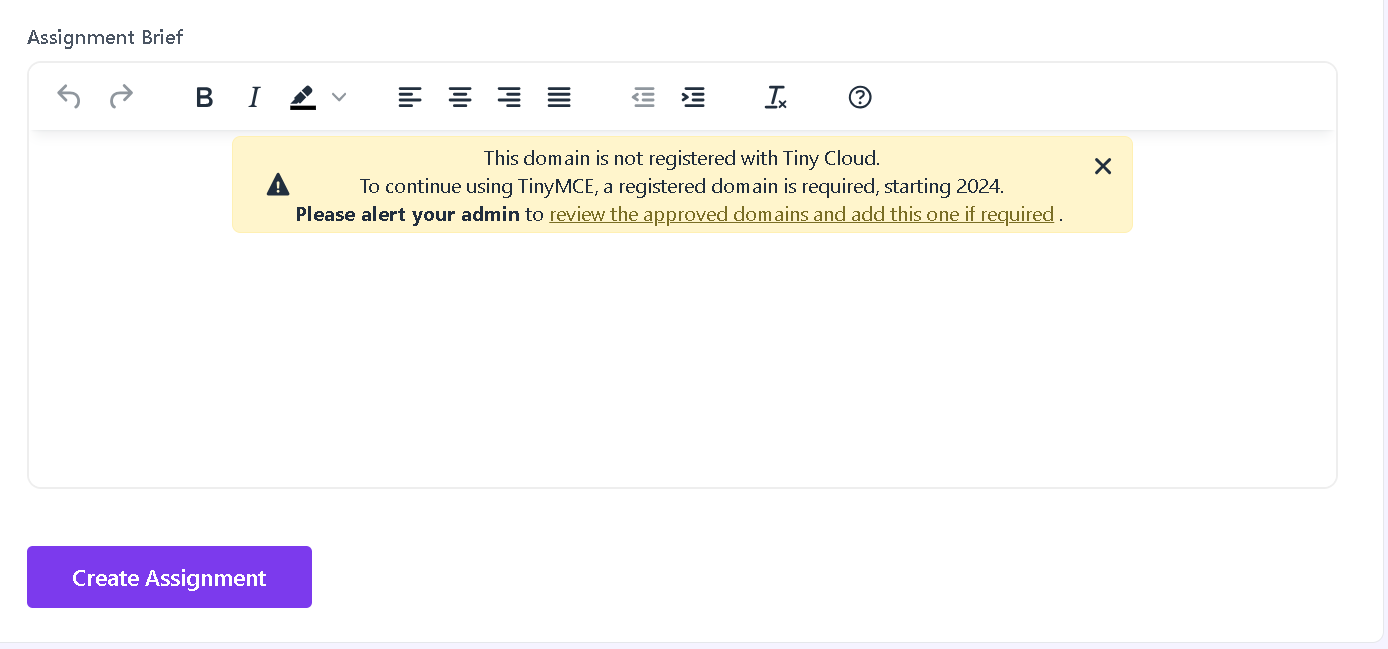
* AssignmentsList - To show all existing assignments
* AddAssignment - To add a new assignment
* SubmittedAssignments - To show assignments students have submitted

1. The AddAssignment component receives the setState function as a prop so that when an assignment is added, it can set the state back to "show" to return to viewing all assignments.

So in summary, it allows easily switching between views to add, show or view submitted assignments with reusable components for each task. The state and props cleanly manage which view is currently active.

**Add Assignment**

****

****

This is a React component for adding a new assignment. Here are the key things it does:

1. Uses state and refs to manage form data and errors, loading state, and a TinyMCE editor reference.

2. Makes a call to a custom hook `useGetAllCourses` to get list of courses to display in the select dropdown.

3. Renders a form with fields for:

- Assignment name

- Selecting a course

- Mark

- Deadline date

- Hints/resources textarea

- Assignment brief WYSIWYG editor

4. Validates the form on submit and shows any errors.

5. On successful submit, makes an API call to create the assignment, shows a toast notification, and sets state back to return to the assignments list view.

6. Handles loading state and errors from the API call.

7. Passes the state setter function down so the parent can set state back to default view after add.

So in summary, it is a reusable assignment creation form that abstracts away the logic for creating assignments, validating input, handling errors and loading states, and integrating with the editor. The parent component controls what view is active while this handles the specific task of adding assignments.

**AssignmentDetails**

This component displays the details of a single assignment. Here's a breakdown:

1. It uses the useLocation hook to access the state data passed to this route, which contains:

- name

- mark

- deadline

- createdAt

- hints

- brief

2. Renders a title and overview info like mark, deadlines, creation date.

3. Conditionally shows the "Assignment Hints/Resources" section if hints data exists.

4. Shows the "Assignment Brief" by dangerously setting the HTML from the brief state.

5. Formats dates and numbers to display nicely.

6. Wraps different sections in semantic HTML elements like <hr> and headings.

So in summary, it takes the assignment data, processes it and displays it in a clean readable layout with good information hierarchy. Helps reuse this data display across the app without duplication. The location state allows easily passing the data to this route.

Let me know if any part needs more explanation!

**Assignment List**

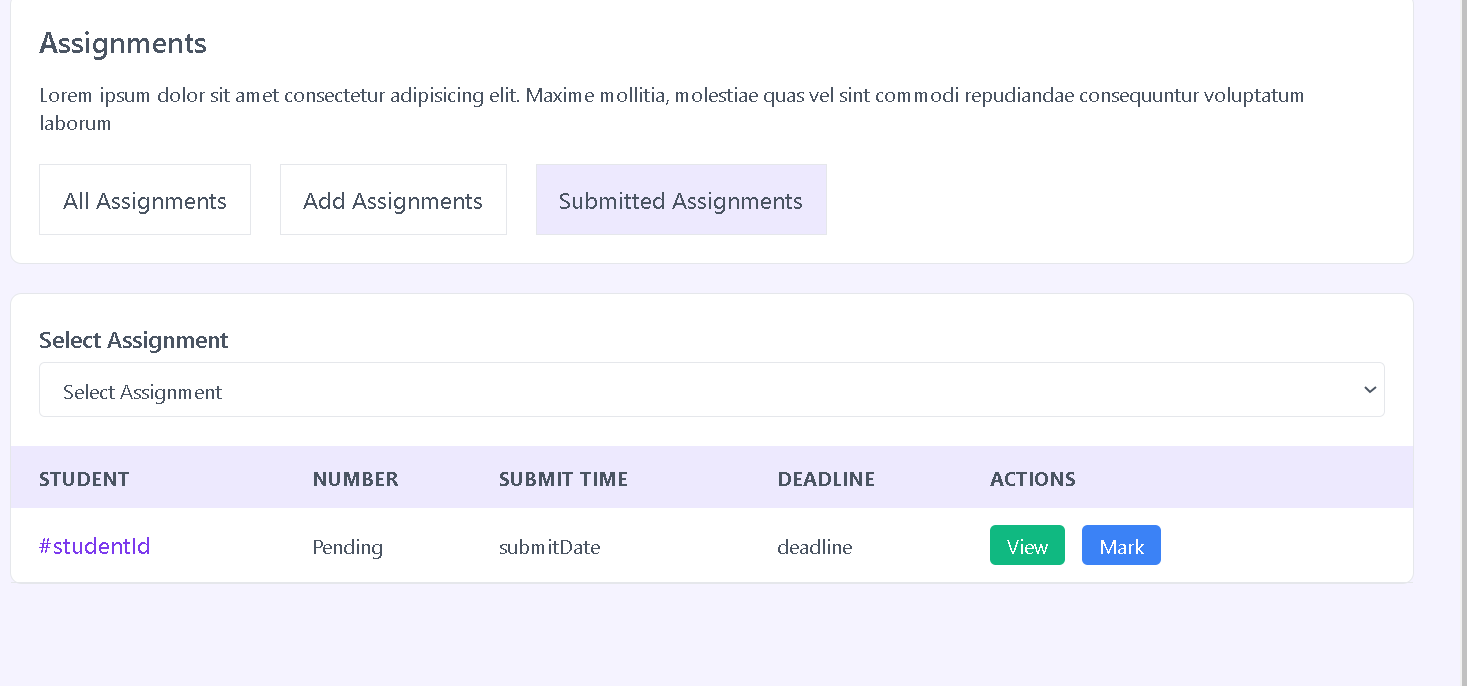
This is a React component to display a list of assignments. Here's what it does:

1. Uses the useGetAllAssignments custom hook to fetch assignments data and loading state.
2. Renders a table with key info columns:
   * Assignment name
   * Course name
   * Number of submissions
   * Deadline
3. Shows a loading skeleton if data is still loading.
4. Maps through the assignments data and displays each one in a table row.
5. The assignment name and course name are clickable buttons to navigate to their detail pages.
6. Passes the assignment data in state to the detail page.
7. Shows a no assignments message if the list is empty.

So in summary:

* It's a reusable table component to display assignments.
* Abstracts away the loading and mapping logic.
* Allows drilling down to more details on click.
* Conditionally shows loading, empty and filled states.

**Submitted Assignments**



This component displays a list of assignments that have been submitted by students. Here is what it does:

1. Renders a select input to allow choosing an assignment to view submissions for.

2. Displays a table with columns for:

- Student name and ID (links to student dashboard)

- Submission number

- Submit date

- Deadline

- Actions (View, Mark buttons)

3. The table data is currently hardcoded, but would be dynamic based on the selected assignment.

4. The View and Mark buttons would open modals/pages to view the submission and add a grade.

5. Shows user and deadline info formatted nicely.

6. Links back to the student dashboard page for more details on a submission.

So in summary, this is a reusable table component to display assignments submissions across different assignments. It keeps things DRY by abstracting out the display and format of the data. The actions allow easy access to grading and evaluating workflow.

Let me know if any additional explanation would be helpful!