Unit Test - TemplateDetail

Test Case 1: Handles Task Addition

- **Objective**: Verify that the system correctly handles the addition of new tasks.
- **Method**: `fireEvent`
- **Test Steps**:
  a. Mock the API call to return an initial state without tasks.
  b. Render the `TemplateDetail` component within a `Router`.
  c. Find and click the "Add Task" button.
  d. Fill in the task name and link inputs.
  e. Check if the inputs reflect the entered values.
- **Expected Result**:
  - The input for the task name should display "Test Task".
  - The input for the link should display "http://newlink.com".

Test Case 2: Updates Tasks on Blur

- **Objective**: Ensure that task updates are handled correctly when an input field is blurred.
- **Method**: `fireEvent`
- **Test Steps**:
  a. Render the `Item` sub-component with initial task details.
  b. Modify the task name input.
  c. Trigger a blur event on the task name input.
  d. Verify that the update function is called with the new task details.
- **Expected Result**:
  - The `onTaskUpdate` function should be called with the updated task name "Updated Task" and the original link.

Test Case 3: Deletes a Task

- **Objective**: Confirm that tasks can be deleted successfully.
- **Method**: `fireEvent`
- **Test Steps**:
  a. Render the `Item` sub-component with a deletable task.
  b. Click the delete button associated with the task.
  c. Verify that the removal function is called for the correct task index.
- **Expected Result**:
  - The `onTaskRemove` function should be called with the index `0`, indicating the first task is targeted for deletion.