**Remove "Loading banner…" text:**  
The "Loading banner…" message is unnecessary. Instead, we have added a background color as a fallback in case the image fails to load. Alternatively, we can implement a skeleton loader to maintain the structure of the UI.

**Prevent page reload on form submission:**  
The form causes the page to reload upon submission. To prevent this, we need to include event.preventDefault(); within the handleSubmit function.

**Reset title input field after form submission:**  
The title input field does not clear automatically after form submission. To fix this, we need to bind the input field to the state by adding value={title} within the input tag, where title is the state variable that holds the current value.

**Add a unique key to the ToDoItem component:**  
In the ToDoItem.tsx file, the <li> element needs a unique key attribute. This can be achieved by adding key={todo.title}, where todo is the object received as a prop and title is a unique field within that object.

**Ensure ToDo item is added to the list on form submission:**  
Currently, the new ToDo item is not being added to the list upon form submission. In index.js, replace the following lines:

**todos.push(newTodo);**

**setTodos(todos);**

with:

**setTodos((prevTodos) => [...prevTodos, newTodo]);**

This ensures that the item is added to the list and displayed in the UI. Additionally, modify the object properties like isCompleted: true and isUrgent: true to reflect the required state changes.