ADVANCED WEB TECHNOLOGY

React Todo List

JUNE 9, 2024

BILAL MEMOOD
ID: 20006105027

1. How to create React project

   Step 1:

```
A:\Eight Semester Data\Advanced web technology\Assignment No 3 React>npm i -g create-react-app
npm WARN deprecated inflight@1.0.6: This module is not supported, and leaks memory. Do not use it. Check out lru-cache i
f you want a good and tested way to coalesce async requests by a key value, which is much more comprehensive and powerfu
l.
npm WARN deprecated rimraf@2.7.1: Rimraf versions prior to v4 are no longer supported
npm WARN deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
npm WARN deprecated uid-number@0.0.6: This package is no longer supported.
npm WARN deprecated fstream-ignore@1.0.5: This package is no longer supported.
npm WARN deprecated fstream@1.0.12: This package is no longer supported.
npm WARN deprecated tar@2.2.2: This version of tar is no longer supported, and will not receive security updates. Please
 upgrade asap.

added 64 packages in 12s

4 packages are looking for funding
  run `npm fund` for details
```

   Step 2

```
A:\Eight Semester Data\Advanced web technology\Assignment No 3 React>create-react-app assignment 3

Creating a new React app in A:\Eight Semester Data\Advanced web technology\Assignment No 3 React\assignment.

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...
```

2. How to start React project

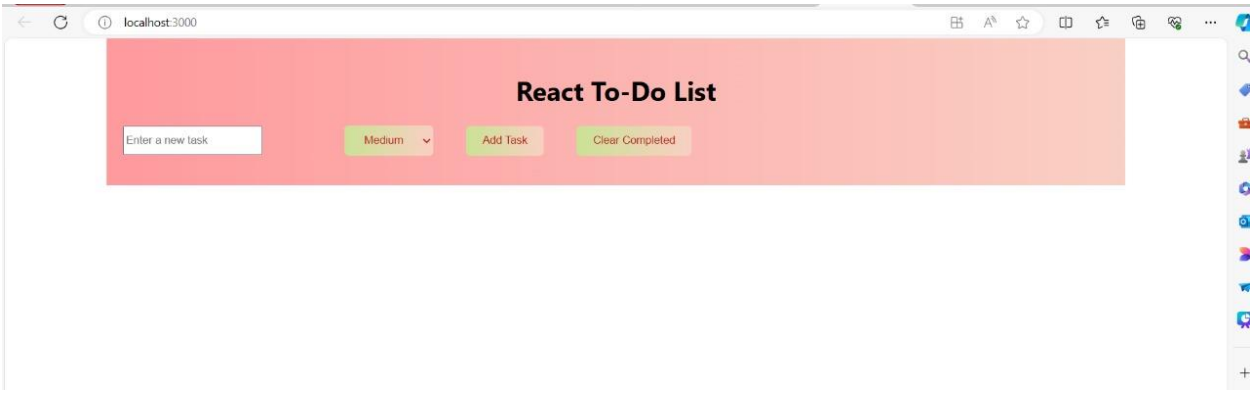
   Cd assignment 3

   npm start

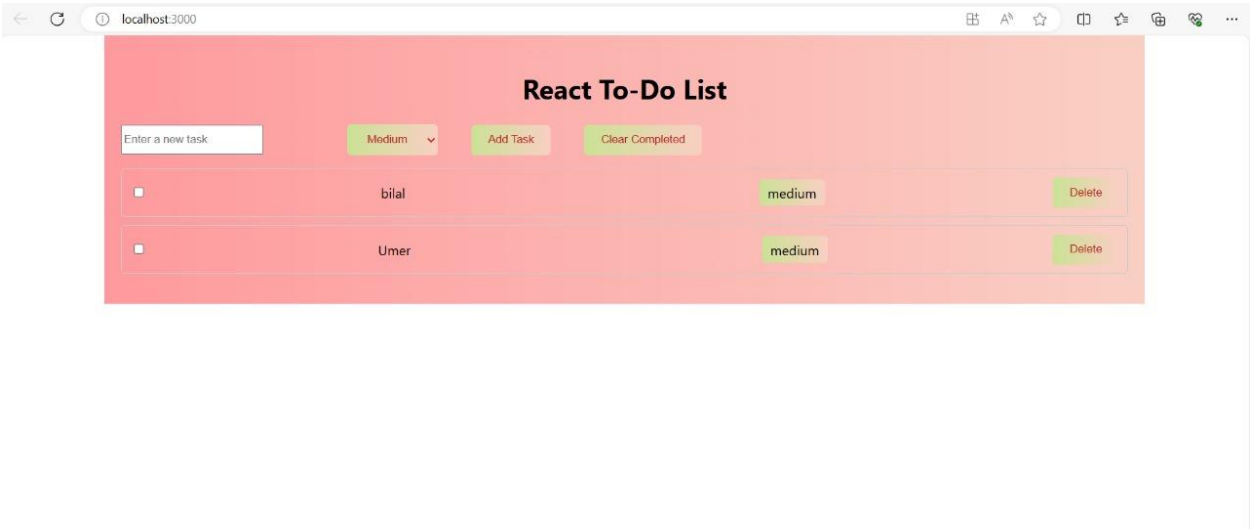
```
A:\Eight Semester Data\Advanced web technology\Assignment No 3 React\assignment>npm start

> assignment@0.1.0 start
> react-scripts start
```

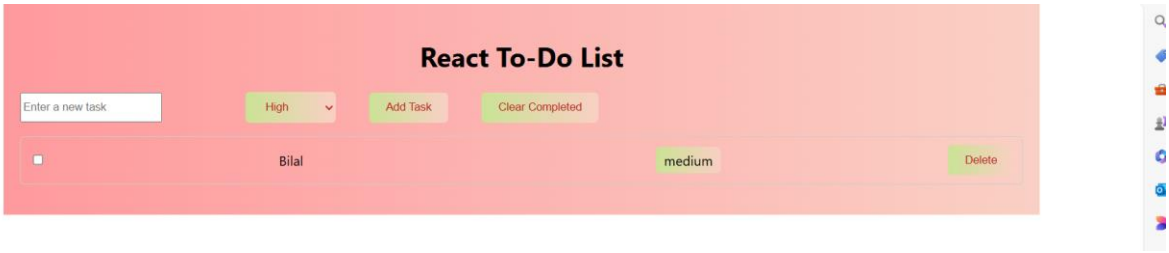
3. Now project is started and show output



4. I created todo list of two entry

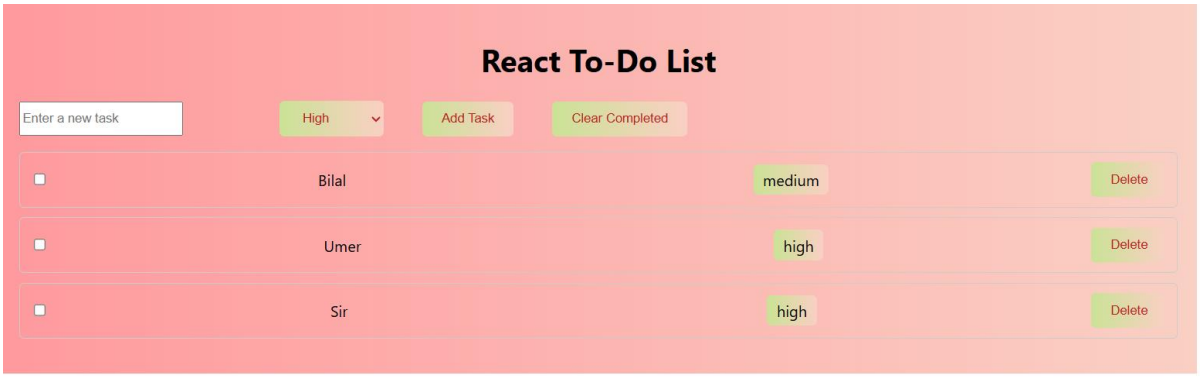


5. User delete unwanted by click on Delete button

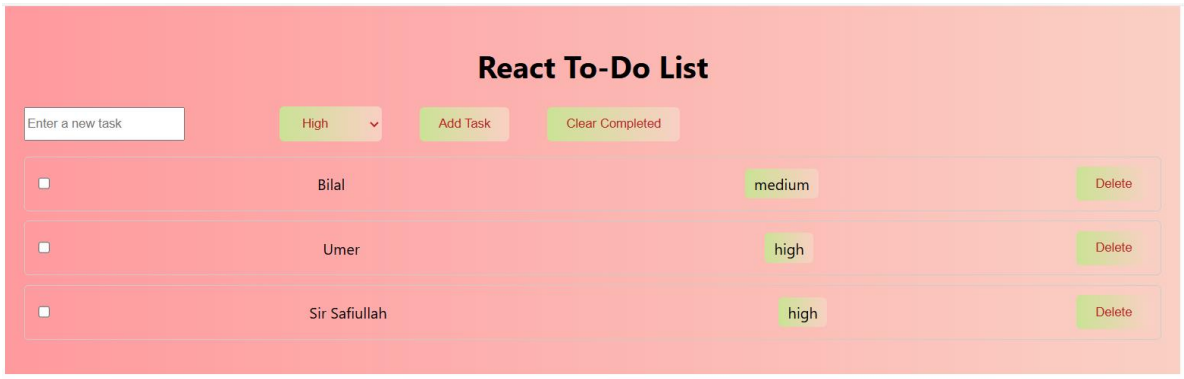


6. Now I am Edit todo list

Step 1 : I create new todo on existing list that name is Sir

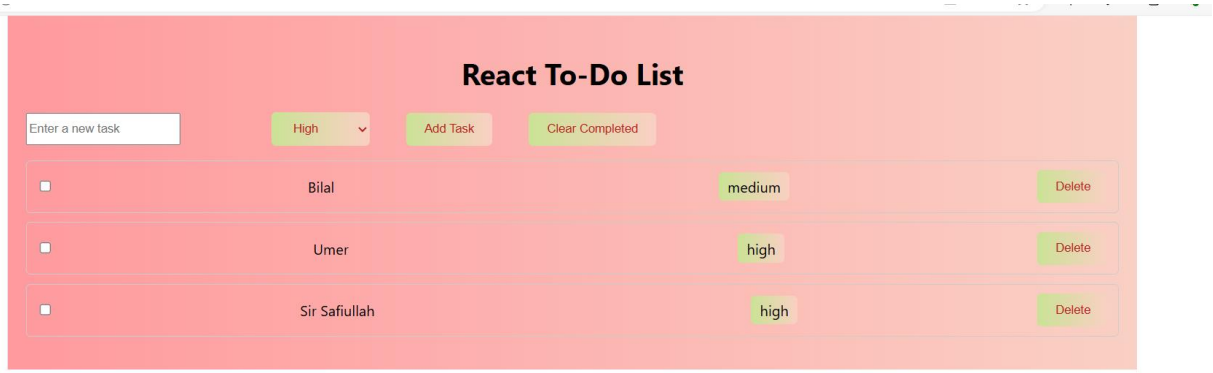


Step 2: Now I updated Sir to Sir Safiullah by double click on that name:

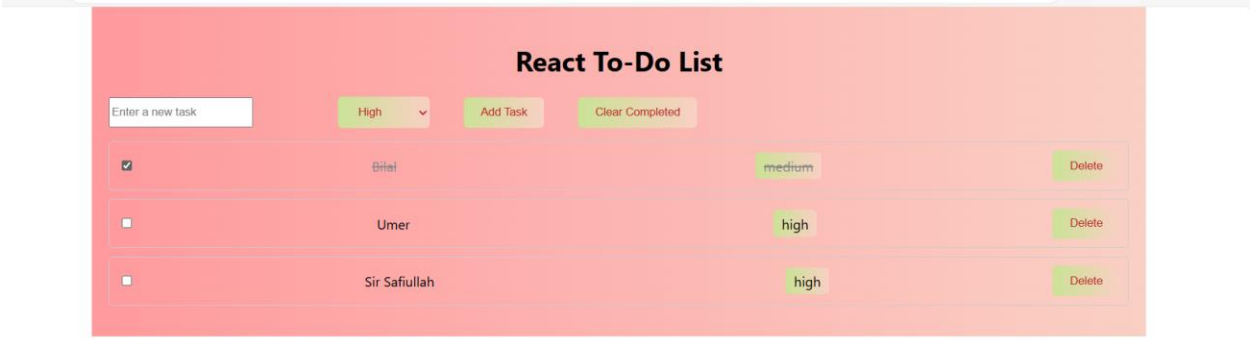


7. And Now Data are stored in local storage

When I refresh the page data are fetch

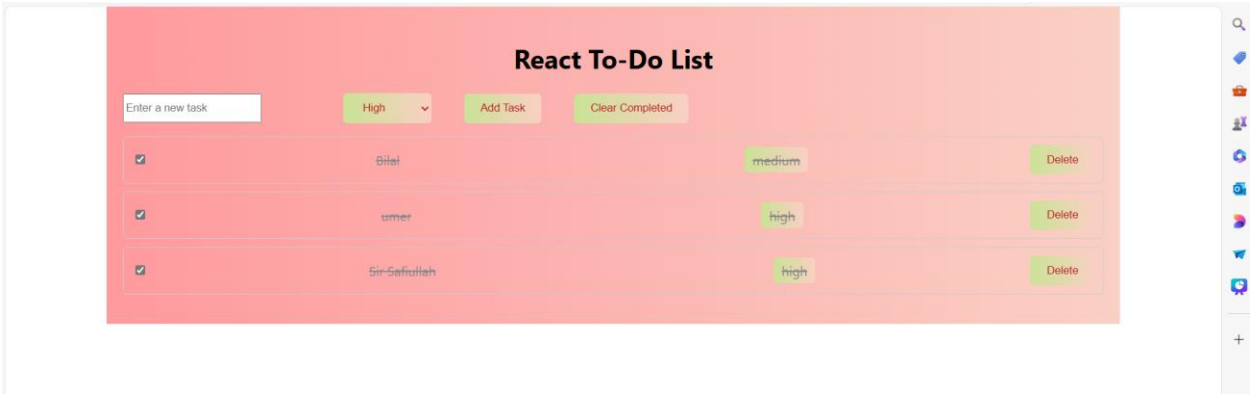


8. And when work are complete I checked that in first entry

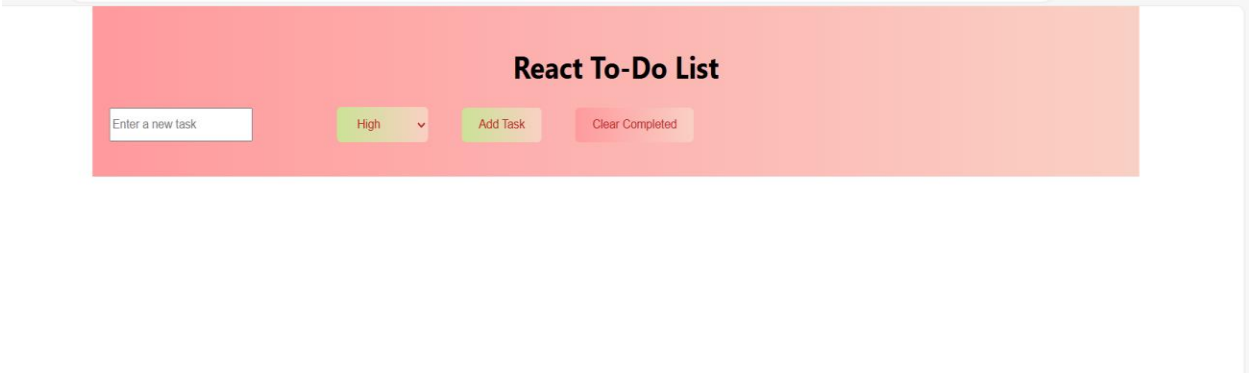


9: Now i delete list of todo using clear completed button

Step 1:



Step 2:



10: App.js

```
import React, { useState, useEffect } from 'react';
```

```
import './App.css';
import TaskItem from './TaskItem';
import { CSSTransition, TransitionGroup } from 'react-transition-group';

function App() {
  const [tasks, setTasks] = useState([]);
  const [newTask, setNewTask] = useState('');
  const [priority, setPriority] = useState('medium');

  useEffect(() => {
    console.log("Fetching tasks from local storage...");
    const savedTasks = JSON.parse(localStorage.getItem('tasks')) || [];
    setTasks(savedTasks);
    setPriority(localStorage.getItem('priority') || 'medium');
  }, []);


  useEffect(() => {
    console.log("Saving tasks to local storage...");
    localStorage.setItem('tasks', JSON.stringify(tasks));
  }, [tasks]);

  useEffect(() => {
    console.log("Saving priority to local storage...");
    localStorage.setItem('priority', priority);
  }, [priority]);


  const addTask = () => {
    if (newTask.trim() === '') return;
    const updatedTasks = [...tasks, { text: newTask, completed: false, priority
}];
    setTasks(updatedTasks);
    setNewTask('');
  };

  const toggleTaskCompletion = index => {
    const updatedTasks = [...tasks];
    updatedTasks[index].completed = !updatedTasks[index].completed;
    setTasks(updatedTasks);
  };

  const deleteTask = index => {
    const updatedTasks = tasks.filter((_, i) => i !== index);
    setTasks(updatedTasks);
```

```
  };

  const clearCompleted = () => {
    const updatedTasks = tasks.filter(task => !task.completed);
    setTasks(updatedTasks);
  };

  const editTask = (index, newText) => {
    const updatedTasks = [...tasks];
    updatedTasks[index].text = newText;
    setTasks(updatedTasks);
  };

  return (
    <div className="App">
      <h1>React To-Do List</h1>
      <input
        type="text"
        value={newTask}
        onChange={e => setNewTask(e.target.value)}
        onKeyDown={e => (e.key === 'Enter' ? addTask() : null)}
        placeholder="Enter a new task"
      />
      <select id = "abc" value={priority} onChange={e =>
setPriority(e.target.value)}>
        <option value="high">High</option>
        <option value="medium">Medium</option>
        <option value="low">Low</option>
      </select>
      <button id = "abc" onClick={addTask}>Add Task</button>
      <button id = "abc" onClick={clearCompleted}>Clear Completed</button>
      <TransitionGroup component="ul">
        {tasks.map((task, index) => (
          <CSSTransition key={index} timeout={300} classNames="fade">
            <TaskItem
              task={task}
              index={index}
              toggleTaskCompletion={toggleTaskCompletion}
              deleteTask={deleteTask}
              editTask={editTask}
            />
          </CSSTransition>
        ))}
      </TransitionGroup>
    </div>
```

```
  );
}

export default App;
```

11: App.css

```
/* Updated CSS with gradient background and button styles */
.App {
  max-width: 1200px;
  max-height: 1500px;
  margin: 0 auto;
  padding: 20px;
  background: linear-gradient(to right, #ff9a9e, #fad0c4); /* Gradient background
*/
}

h1 {
  text-align: center;
}

ul {
  list-style-type: none;
  padding: 0;
}

li {
  display: flex;
  align-items: center;
  justify-content: space-between;
  margin-bottom: 10px;
  padding: 10px;
  border: 1px solid #ccc;
  border-radius: 5px;
}

.completed span {
  text-decoration: line-through;
  color: #888;
}

input[type="checkbox"] {
  margin-right: 10px;
```

```
}

input[type="text"] {
  flex: 1;
  margin-right: 100px;
  height: 30px;
  width:150;

}

.priority {
  margin-left: 150px;
  padding: 5px 10px;
  border-radius: 5px;
  background: linear-gradient(to right, #cbe396, #fad0c4);
}

.priority.high {
  background-color: #ff6b6b;
}

.priority.medium {
  background-color: #ffd166;
}

.priority.low {
  background-color: #a0dfed;
}

/* Button Styles */
#abc {
  background: linear-gradient(to right, #cbe396, #fad0c4); /* Gradient button
color */
  color: #bd2222;
  padding: 10px 20px;

  border: none;
  border-radius: 5px;
  cursor: pointer;

}

#abc:hover {
  background: linear-gradient(to right, #ff9a9e, #fad0c4); /* Gradient button
hover color */
```

```
}

/* Add space between buttons */
#abc + #abc {
  margin-left: 40px;
}
```

12: TaskItem.js

```
import React, { useState } from 'react';
import { CSSTransition } from 'react-transition-group';
import classnames from 'classnames';

function TaskItem({ task, index, toggleTaskCompletion, deleteTask, editTask }) {
  const [isEditing, setIsEditing] = useState(false);
  const [editText, setEditText] = useState(task.text);

  const handleEdit = () => {
    if (isEditing && editText.trim() !== '') {
      editTask(index, editText);
      setIsEditing(false);
    }
  };

  const handleKeyDown = e => {
    if (e.key === 'Enter') {
      handleEdit();
    } else if (e.key === 'Escape') {
      setEditText(task.text);
      setIsEditing(false);
    }
  };

  return (
    <li className={classnames({ 'completed': task.completed })}>
      <input
        type="checkbox"
        checked={task.completed}
        onChange={() => toggleTaskCompletion(index)}
      />
      {isEditing ? (
        <input
          type="text"
          value={editText}
          onChange={e => setEditText(e.target.value)}
```

```
          onBlur={handleEdit}
          onKeyDown={handleKeyDown}
        />
      ) : (
        <span onDoubleClick={() => setIsEditing(true)}>{task.text}</span>
      )}
      <span  className={`priority ${task.priority}`}>{task.priority}</span>
      <button id = "abc" onClick={() => deleteTask(index)}>Delete</button>
    </li>
  );
}

export default TaskItem;
```