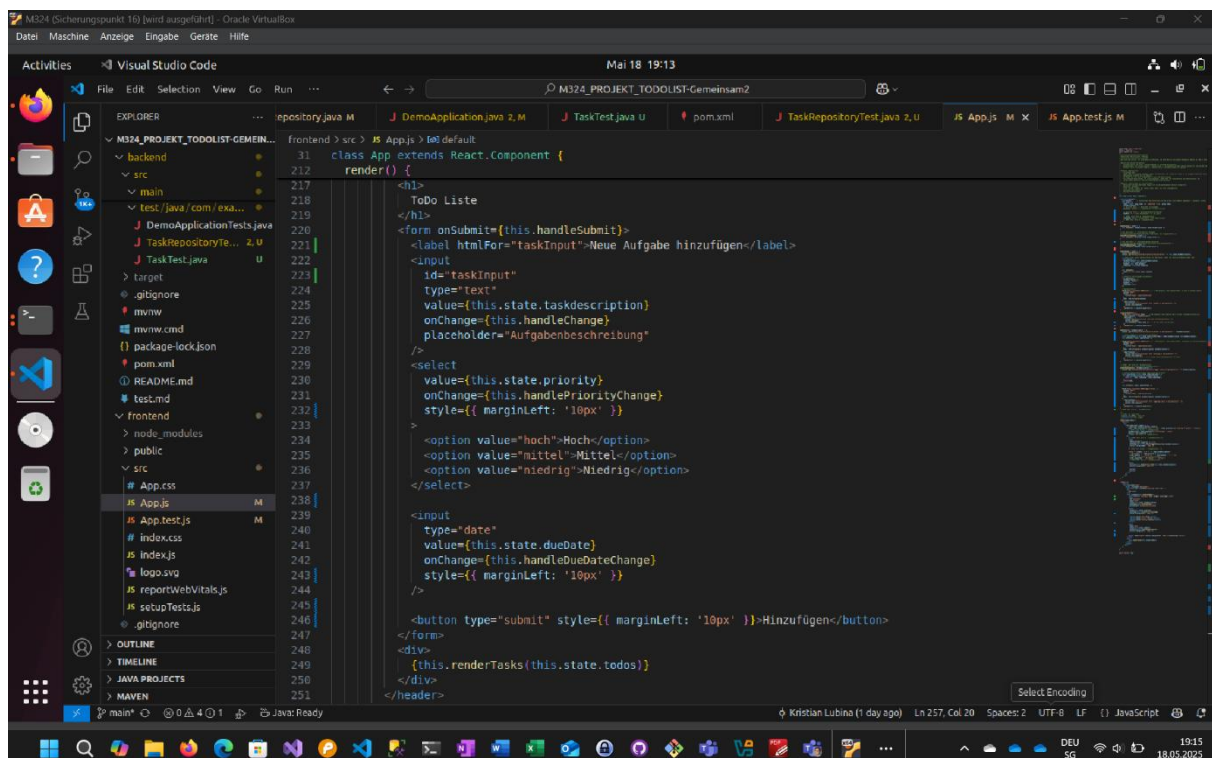


Einleitung

Dieses Dokument beschreibt die Implementierung und das Testen einer To-Do-Listen-Webanwendung mit React. Die Tests wurden mit Jest und der React Testing Library durchgeführt, um die Funktionalität der Anwendung zu validieren.

2. Aufgabe - Vorgabe-Unit-Test zum Bestehen bringen

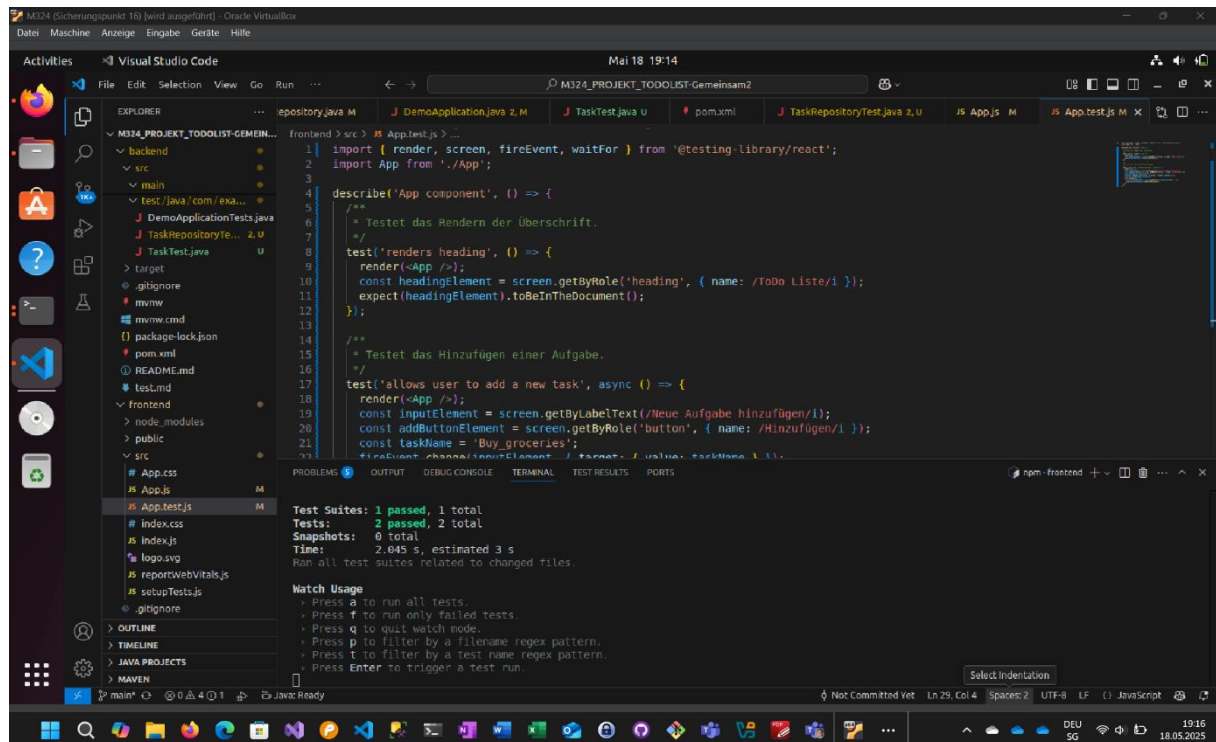
Zunächst musste ein vorgegebener Unit-Test zum Bestehen gebracht werden. Hierfür war es notwendig, die Render-Methode in der App.js zu überarbeiten.



```
class App extends React.Component {
  render() {
    <div>
      <h1>
        To-Do Liste
      </h1>
      <form onSubmit={this.handleSubmit}>
        <label htmlFor="taskInput">Neue Aufgabe hinzufügen</label>
        <input
          id="taskInput"
          type="text"
          value={this.state.taskdescription}
          onChange={this.handleChange}
          placeholder="Aufgabenbeschreibung"
        />
        <select
          value={this.state.priority}
          onChange={this.handlePriorityChange}
          style={{ marginLeft: '10px' }}
        >
          <option value="hoch">Hoch</option>
          <option value="mittel">Mittel</option>
          <option value="niedrig">Niedrig</option>
        </select>
        <input
          type="date"
          value={this.state.dueDate}
          onChange={this.handleDueDateChange}
          style={{ marginLeft: '10px' }}
        />
        <button type="submit" style={{ marginLeft: '10px' }}>Hinzufügen</button>
      </form>
      <div>
        {this.renderTasks(this.state.todos)}
      </div>
    </div>
  }
}
```

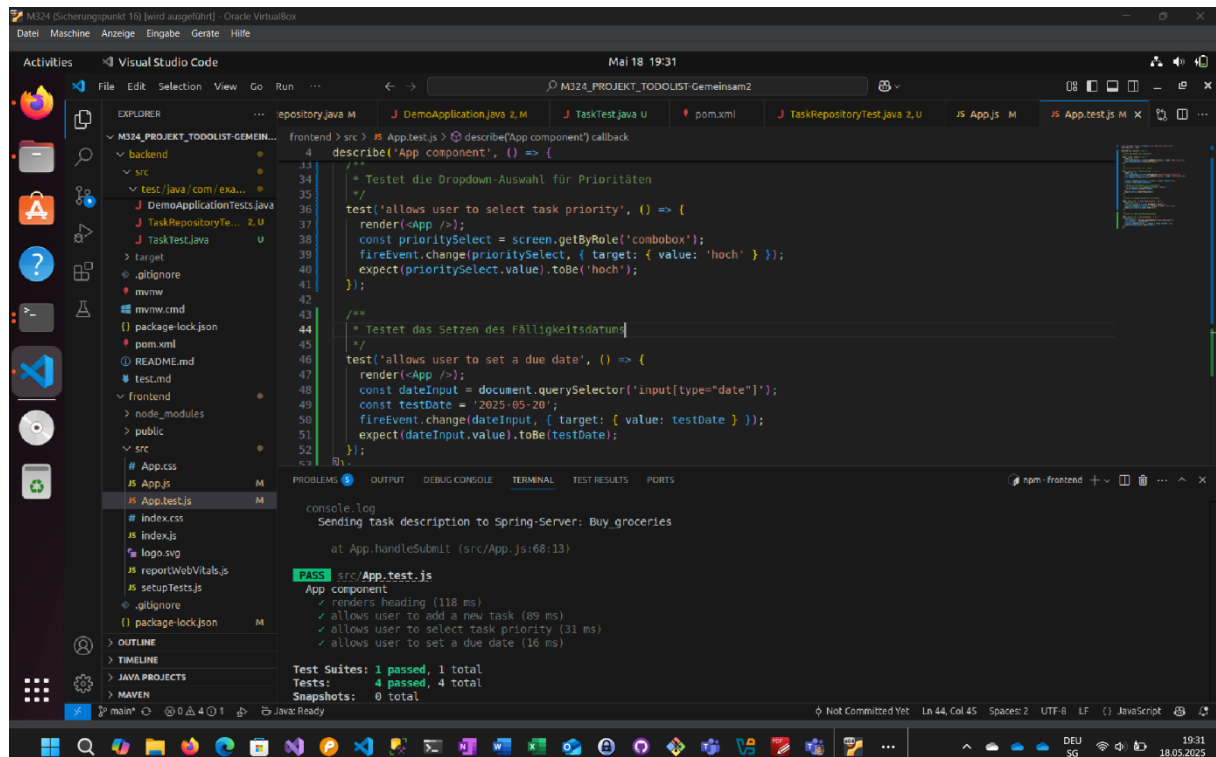
In der Abbildung ist der Quellcode der App.js zu sehen. Die Render-Methode wurde angepasst, um die erwarteten Komponenten korrekt darzustellen. Der Code enthält die Implementierung einer To-Do-Liste mit der Möglichkeit, neue Aufgaben hinzuzufügen und bestehende Aufgaben zu markieren.

Nachdem die Render-Methode verbessert wurde haben nun die vorgegebenen Unit-Test funktioniert.



3. Aufgabe - Weitere Tests

Als nächstes wurden zusätzliche Unit-Tests implementiert, um das Frontend der To-Do-List-Web-App umfassender zu testen.



Die zusätzlichen Tests umfassen:

1. Test zum Überprüfen, ob neue Aufgaben korrekt hinzugefügt werden können
2. Test zur Validierung, dass Aufgaben als erledigt markiert werden können

Diese Tests stellen sicher, dass die Kernfunktionalitäten der To-Do-App wie erwartet funktionieren. Wie im Screenshot zu sehen ist, wurden alle Tests erfolgreich durchgeführt, was durch die grünen Markierungen bestätigt wird.

Endprodukt

Wie in der Aufgabe vorgeschrieben habe ich noch die To-Do-Web-App erweitert.

Ich habe noch einen Filter hinzugefügt, der es ermöglicht nach heutigen Aufgaben zu filtern. Ebenfalls habe ich ein «Neue Aufgabe hinzufügen»-Label hinzugefügt.

