

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4   <head>
5     <meta charset="utf-8">
6     <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
7     <script src="https://unpkg.com/axios/dist/axios.min.js"></script>
8     <title>Assignment 2</title>
9   </head>
10  <body>
11    <div id="app">
12      <h1>Movies Application</h1>
13      <div>
14        <button v-on:click="handleCreateTable">Create Table</button>
15      </div>
16      <div v-if="tableCreatedError">
17        <p>{{tableCreatedErrorMessage}}</p>
18      </div>
19      <div v-if="tableCreated">
20        <p>{{tableCreatedMessage}}</p>
21      </div>
22      <br/>
23      <form @submit.prevent="handleQuery">
24        <label>Movie Name
25          <input type="text" v-model="queryTitle">
26        </label>
27        <label>Movie year
28          <input type="text" v-model="queryYear">
29        </label>
30        <button v-on:submit.prevent>Submit</button>
31      </form>
32      <br/>
33      <div v-if="hasQueryResponse && !queryError">
34        <table border=1>
35          <tr>
36            <th>
37              Movie Name
38            </th>
39            <th>
40              Movie year
41            </th>
42            <th>
43              Release Date
44            </th>
45            <th>
46              Rank
47            </th>
48          </tr>
49          <tr v-for="movie in movies" :key="movie.title">
50            <td>
51              {{movie.title}}
52            </td>
53            <td>
54              {{movie.year}}
55            </td>
56            <td>
57              {{movie.release_date}}
58            </td>
59            <td>
60              {{movie.rank}}

```

```
61         </td>
62     </table>
63     <br/>
64 </div>
65 <div v-if="queryError">
66     <p>{{queryErrorMessage}}</p>
67 </div>
68 <div>
69     <button v-on:click="handleDeleteTable">Delete Table</button>
70 </div>
71 <div v-if="tableDeleted">
72     <p>{{tableDeletedMessage}}</p>
73 </div>
74 <div v-if="tableDeletedError">
75     <p>{{tableDeletedErrorMessage}}</p>
76 </div>
77
78
79 </div>
80 </body>
81
82
83 <script>
84     var app = new Vue({
85         el: '#app',
86         data: {
87             movies: [],
88             tableCreatedError: false,
89             tableCreatedErrorMessage: "",
90             tableCreated: false,
91             tableCreatedMessage: "Table Created",
92             queryYear: null,
93             queryTitle: null,
94             queryError: false,
95             queryErrorMessage: "",
96             hasQueryResponse: false,
97             tableDeleted: false,
98             tableDeletedMessage: "Table Deleted",
99             tableDeletedError: false,
100             tableDeletedErrorMessage: "Error deleting table"
101         },
102         methods: {
103             resetErrorsAndSuccessMessages(){
104                 this.hasQueryResponse = false
105                 this.queryError = false
106                 this.tableCreated = false
107                 this.tableCreatedError = false
108                 this.tableDeleted = false
109                 this.tableDeletedError = false
110             },
111             async handleCreateTable() {
112
113                 this.resetErrorsAndSuccessMessages()
114
115                 const response = await fetch("api/create-database")
116
117                 if(response.status >=200 && response.status <=299){
118                     this.tableCreated = true
119                 }else{
```

```
121         const body = await response.json();
122         this.tableCreatedError = true
123         this.tableCreatedErrorMessage = body.message
124     }
125 },
126 },
127 },
128 async handleDeleteTable() {
129     this.resetErrorsAndSuccessMessages()
130     const response = await fetch("api/delete-database")
131     if(response.status >=200 && response.status <=299){
132         this.tableDeleted = true
133     }else{
134         const body = await response.json();
135         this.tableDeletedError = true
136         this.tableDeletedErrorMessage = body.message
137     }
138 },
139 },
140 },
141 },
142 },
143 },
144 async handleQuery() {
145     this.resetErrorsAndSuccessMessages()
146     if(!this.queryTitle || !this.queryYear){
147         this.queryError = true
148         this.queryErrorMessage = "Please enter a year and
149 title"
150         console.log('hit error')
151     }else{
152         const response = await fetch(`api/get-movies?
153 title=${this.queryTitle}&year=${this.queryYear}`)
154         if(response.status >=200 && response.status <=299){
155             const body = await response.json();
156             this.movies = body
157             if(this.movies.length === 0){
158                 this.hasQueryResponse = false
159                 this.queryError = true
160                 this.queryErrorMessage = "No movies found
161 matching this query"
162             }else{
163                 this.hasQueryResponse = true
164             }
165         }else{
166             this.queryError = true
167             this.queryErrorMessage = body.message
168             console.log(response)
169         }
170     }
171 },
172 },
173 },
174 },
175 })
176 </script>
177 </html>
```