

index.html

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset="utf-8">
```

```
<title>EmberListView</title>
```

```
<meta name="description" content="">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

```
{{content-for "head"}}
```

```
<link integrity="" rel="stylesheet" href="{{rootURL}}assets/vendor.css">
```

```
<link integrity="" rel="stylesheet" href="{{rootURL}}assets/ember-list-view.css">
```

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.7.2/css/all.min.css" integrity="sha512-Evv84Mr4kqVGRNSglGL/F/aIDqQb7xQ2vcrdlwxfjThSH8CSR7PBEakCr51Ck+w+/U6swU2Im1vVX0SVk9ABhg==" crossorigin="anonymous" referrerpolicy="no-referrer" />
```

```
{{content-for "head-footer"}}
```

```
</head>
```

```
<body>
```

```
{{content-for "body"}}
```

```
<script src="{{rootURL}}assets/vendor.js"></script>
```

```
<script src="{{rootURL}}assets/ember-list-view.js"></script>
```

```
{{content-for "body-footer"}}
```

```
</body>
```

```
</html>
```

templates/application.hbs

```
{{#if this.isOverlayVisible}}
<div id="overlay"></div>
{{/if}}

<div class="container">
  <h4>STUDENT LIST</h4>
  <div class="searchDlt">
    <div class="search-cont">
      <i class="fa-solid fa-magnifying-glass"></i>
      <input type="text" id="search" placeholder="Search here..." {{on "keyup"
this.searchTable}}>
    </div>
    <div>
      <button {{on "click" this.bulkDelete}} class="btn-cd dlt"><i class="fa-solid fa-list-
check"></i>Bulk Delete</button>
      <button {{on "click" this.openForm}} class="btn-cd"><i class="fa-solid fa-user-
plus"></i>Add Student</button>
    </div>
  </div>

  <StudentList @students={{this.students}} @deleteRow={{this.deleteRow}} />
</div>

{{#if this.isFormVisible}}
<StudentForm @addRow={{this.addRow}} @closeForm={{this.closeForm}} />
{{/if}}
```

components/student-form.hbs

```
<div id="formCont">

  <span id="close-btn" {{on "click" @closeForm}}><i class="fa-solid fa-circle-
xmark"></i></span>

  <input type="text" {{on "input" this.updateName}} placeholder="Enter Name">

  <input type="text" {{on "input" this.updateReg}} placeholder="Enter Reg No">

  <input type="text" {{on "input" this.updateDept}} placeholder="Enter Department">

  <input type="text" {{on "input" this.updateClg}} placeholder="Enter College">

  <div class="decision">

    <button {{on "click" @closeForm}} class="btn-cd cancel">Cancel</button>

    <button {{on "click" this.createStudent}} class="btn-cd">Create</button>

  </div>

</div>
```

components/student-list.hbs

```
<div class="table-container">

  <table id="table">

    <tr>

      <th>Select</th>

      <th>Name</th>

      <th>Reg No</th>

      <th>Department</th>

      <th>College</th>

      <th>Action</th>

    </tr>

    {{#each @students as |student|}}

      <tr>

        <td><input type="checkbox"></td>
```

```

        <td>{{student.name}}</td>

        <td>{{student.reg}}</td>

        <td>{{student.dept}}</td>

        <td>{{student.clg}}</td>

        <td><button {{on "click" (fn @deleteRow student.reg)}}><i class="fa-solid fa-
trash"></i>Delete</button></td>

    </tr>

    {{/each}}

</table>

</div>

```

components/student-form.js

```

import Component from '@glimmer/component';
import { action } from '@ember/object';
import { tracked } from '@glimmer/tracking';

export default class StudentFormComponent extends Component {

    @tracked name = '';

    @tracked reg = '';

    @tracked dept = '';

    @tracked clg = '';

    @action updateName(event) {
        this.name = event.target.value;
    }

    @action updateReg(event) {
        this.reg = event.target.value;
    }

```

```
}
```

```
@action updateDept(event) {  
    this.dept = event.target.value;  
}
```

```
@action updateClg(event) {  
    this.clg = event.target.value;  
}
```

```
@action createStudent() {  
  
    if(!this.name || !this.reg || !this.dept || !this.clg) {  
        alert("Fill all the rows");  
        return;  
    }
```

```
    this.args.addRow({  
        name: this.name,  
        reg: this.reg,  
        dept: this.dept,  
        clg: this.clg  
    });
```

```
    this.args.closeForm();  
}  
}
```

controllers/application.js

```
import Controller from '@ember/controller';
```

```
import { action } from '@ember/object';
```

```
import { tracked } from '@glimmer/tracking';
```

```
export default class ApplicationController extends Controller {
```

```
  @tracked students = [
```

```
    {
```

```
      name: "Muhamad",
```

```
      reg: 962821205034,
```

```
      dept: "IT",
```

```
      clg: "UCEN"
```

```
    },
```

```
    {
```

```
      name: "Abishek",
```

```
      reg: 962821205003,
```

```
      dept: "IT",
```

```
      clg: "UCEN"
```

```
    },
```

```
    {
```

```
      name: "Aswin",
```

```
      reg: 962821205016,
```

```
      dept: "IT",
```

```
      clg: "UCEN"
```

```
    },
```

```
    {
```

```
      name: "Ruban",
```

```
    reg: 962821205047,  
    dept: "CSE",  
    clg: "UCEN"  
  },  
  {  
    name: "Surya",  
    reg: 962821205038,  
    dept: "CSE",  
    clg: "UCEN"  
  },  
  {  
    name: "Nithin",  
    reg: 962821205037,  
    dept: "IT",  
    clg: "UCEN"  
  },  
  {  
    name: "Jees",  
    reg: 962821205050,  
    dept: "IT",  
    clg: "UCEN"  
  },  
  {  
    name: "Nithish",  
    reg: 962821205040,  
    dept: "IT",  
    clg: "UCEN"  
  },
```

```
{
  name: "Jones",
  reg: 962821205008,
  dept: "IT",
  clg: "UCEN"
},
];
```

```
@tracked isFormVisible = false;
@tracked isOverlayVisible = false;
```

```
@action addRow(student) {
  this.students = [...this.students, student];
}
```

```
@action deleteRow(reg) {
  this.students = this.students.filter(student => student.reg !== reg);
}
```

```
@action bulkDelete() {
  let checkboxes = document.querySelectorAll("#table input[type='checkbox']:checked");
  if (!checkboxes.length) {
    alert("Select one or more rows to bulk delete");
    return;
  }
```

```
checkboxes.forEach(checkbox => {
  let row = checkbox.parentElement.parentElement;
```



```
        this.deleteRow(row.cells[2].textContent);  
        row.remove();  
    });  
}
```

```
@action searchTable(event) {  
    let input = event.target.value.toLowerCase();  
    let rows = document.querySelectorAll("#table tr");  
  
    rows.forEach((row, index) => {  
        if (index === 0) return;  
        row.style.display = row.innerText.toLowerCase().includes(input) ? "" : "none";  
    });  
}
```

```
@action openForm() {  
    this.isFormVisible = true;  
    this.isOverlayVisible = true;  
}
```

```
@action closeForm() {  
    this.isFormVisible = false;  
    this.isOverlayVisible = false;  
}  
}
```

styles/app.css

/* Ember supports plain CSS out of the box. More info:
<https://cli.emberjs.com/release/advanced-use/stylesheets/> */

```
body {  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  height: 90vh;  
  font-family: Arial, Helvetica, sans-serif;  
}
```

```
#overlay {  
  position: fixed;  
  top: 0;  
  left: 0;  
  height: 100%;  
  width: 100%;  
  background: rgba(0, 0, 0, 0.5);  
  z-index: 1000;  
}
```

```
.container {  
  display: flex;  
  align-items: center;  
  flex-direction: column;  
  padding: 2rem;  
  background-color: #fff;
```

```
border-radius: 10px;
width: 70%;
height: 75vh;
position: relative;
}
```

```
#formCont,table {
margin: 2rem;
}
```

```
#formCont {
display: flex;
flex-direction: column;
gap: 2rem;
position: absolute;
background-color: white;
padding: 3rem;
border-radius: 10px;
width: 30%;
z-index: 1001;
}
```

```
.decision {
display: flex;
justify-content: flex-end;
}
```

```
#close-btn {
```

```
    cursor: pointer;
    position: absolute;
    top: 2rem;
    right: 2rem;
    color: #3e3e3e;
    font-size: 20px;
}
```

```
input {
    margin-right: 1rem;
    padding: 0.5rem 1rem;
    border: none;
    border-bottom: 1px solid #868686;
    background-color: #ffffff;
    outline: none;
}
```

```
input[type=text]:focus {
    border: none;
    border-bottom: 1px solid #868686;
}
```

```
input[type=checkbox] {
    cursor: pointer;
}
```

```
.table-container {
    height: auto;
```

```
    overflow-y: scroll;

    scrollbar-width: none;

    box-shadow: 3px 2px 10px #d3d3d3;

    margin: 1.5rem;

    width: 80%;

    border-radius: 10px;
}
```

```
table {

    background-color: #fff;

    border-radius: 10px;

    border-collapse: collapse;

    color: #3e3e3e;

    margin: 0;

    width: 100%;

}
```

```
td,th {

    padding: 1rem;

    font-size: 14px;

    text-align: center;

}
```

```
th {

    position: sticky;

    top: 0;

    border-bottom: 1px solid #ccc;

    background-color: #f6f6f6;
```

```
}
```

```
.btn-cd {  
    padding: 0.5rem 1rem;  
    border-radius: 20px;  
    border: none;  
    background-color: #4f8091;  
    color: white;  
    cursor: pointer;  
    font-size: 14px;  
    margin-right: 10px;  
}
```

```
.btn-cd i {  
    font-size: 12px;  
}
```

```
i {  
    margin-right: 0.5rem;  
}
```

```
.dlt {  
    background-color: #ffdcdd;  
    color: #d84949;  
}
```

```
.cancel {  
    background-color: white;
```

```
    color: black;
    border: 1px solid #9f9f9f;
}
```

```
button {
    background: transparent;
    border: none;
    cursor: pointer;
    color: #d84949;
}
```

```
button i {
    font-size: 12px;
}
```

```
.searchDlt {
    width: 80%;
    display: flex;
    justify-content: space-between;
    align-items: center;
    margin: 1rem;
    gap: 1rem;
}
```

```
#search {
    padding: 10px 0px;
    border: none;
    width: 100%;
```

```
}
```

```
.search-cont {  
  display: flex;  
  align-items: center;  
  border: 1px solid #ccc;  
  width: 40%;  
  color: #878787;  
  padding: 0 1rem;  
  border-radius: 20px;  
}
```

```
.search-cont i {  
  font-size: 14px;  
}
```

package.json

```
"ember-auto-import": "^2.10.0"  
"ember-cli": "~5.9.0",  
"ember-cli-app-version": "^7.0
```

```
"ember-source": "~5.9.0",  
"ember-template-imports": "^2
```