

## OUTPUT:

### ember-vertical-collection

```
<VerticalCollection @estimateHeight={{50}} @items={{@students}} @key="reg"
  @containerSelector=".table-container" 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 class="btn-ed edit" type="button" {{on "click"
(fn @editRow student.reg)}}>
        <i class="fa-solid fa-pen-to-square"></i>
      </button>
      <button class="btn-ed delete" type="button" {{on
"click" (fn @deleteRow student.reg)}}>
        <i class="fa-solid fa-trash"></i>
      </button>
    </td>
  </tr>
</VerticalCollection>
```

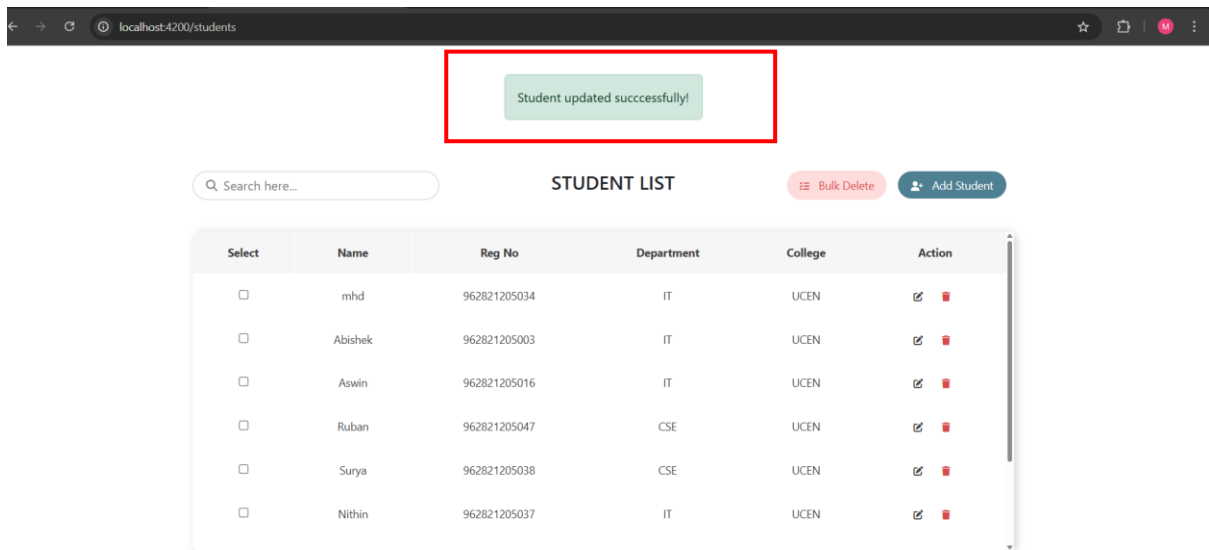


STUDENT LIST

[Bulk Delete](#) [Add Student](#)

Select	Name	Reg No	Department	College	Action
<input type="checkbox"/>	Muhamad	962821205034	IT	UCEN	<a href="#">✎</a> <a href="#">✖</a>
<input type="checkbox"/>	Abishek	962821205003	IT	UCEN	<a href="#">✎</a> <a href="#">✖</a>
<input type="checkbox"/>	Aswin	962821205016	IT	UCEN	<a href="#">✎</a> <a href="#">✖</a>
<input type="checkbox"/>	Ruban	962821205047	CSE	UCEN	<a href="#">✎</a> <a href="#">✖</a>
<input type="checkbox"/>	Surya	962821205038	CSE	UCEN	<a href="#">✎</a> <a href="#">✖</a>
<input type="checkbox"/>	Nithin	962821205037	IT	UCEN	<a href="#">✎</a> <a href="#">✖</a>

## ember-cli-flash



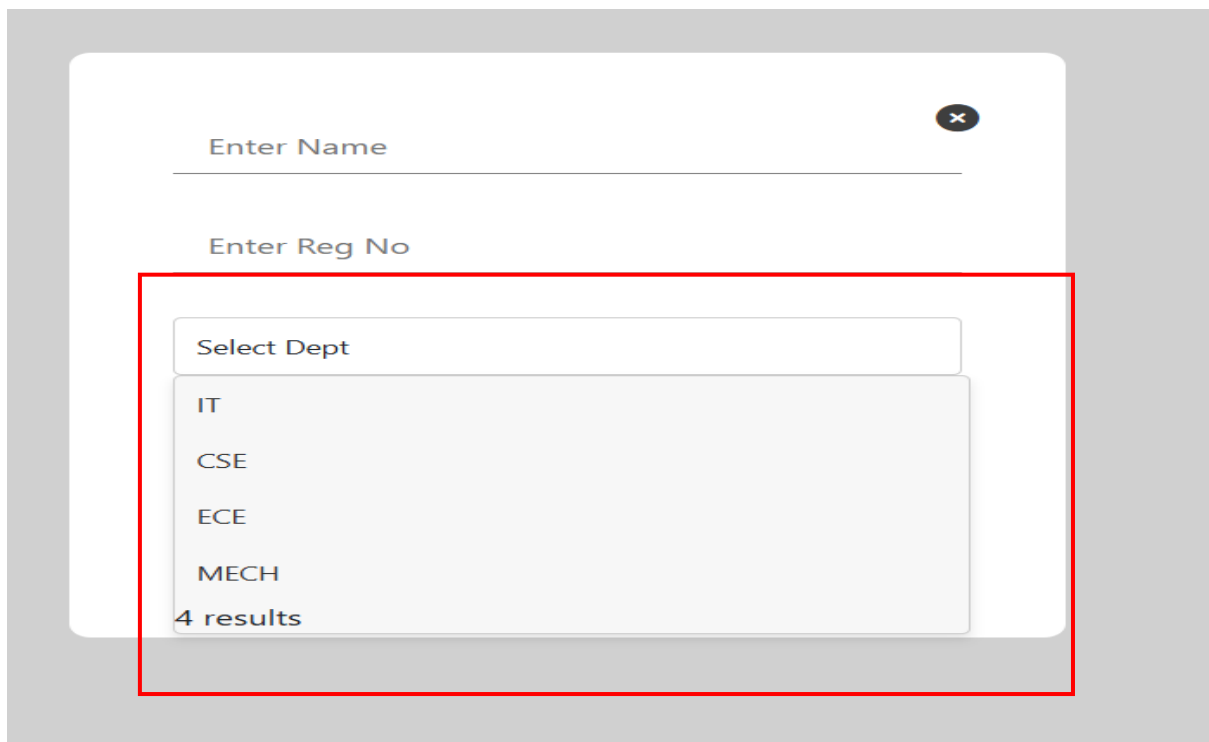
## ember-click-outside

```
<div {{on-click-outside this.closeForm}}>
  <StudentForm @addRow={{this.addRow}} @closeForm={{this.closeForm}} />
</div>
```

## ember-concurrency

```
@task
*searchTable(event) {
  let input = event.target.value.toLowerCase();
  yield timeout(300);
  this.students.searchStudents(input);
}
```

## ember-power-select



## ember-truth-helpers

```
{{#if (and @students (gt (get @students "length") 0)))}}
```

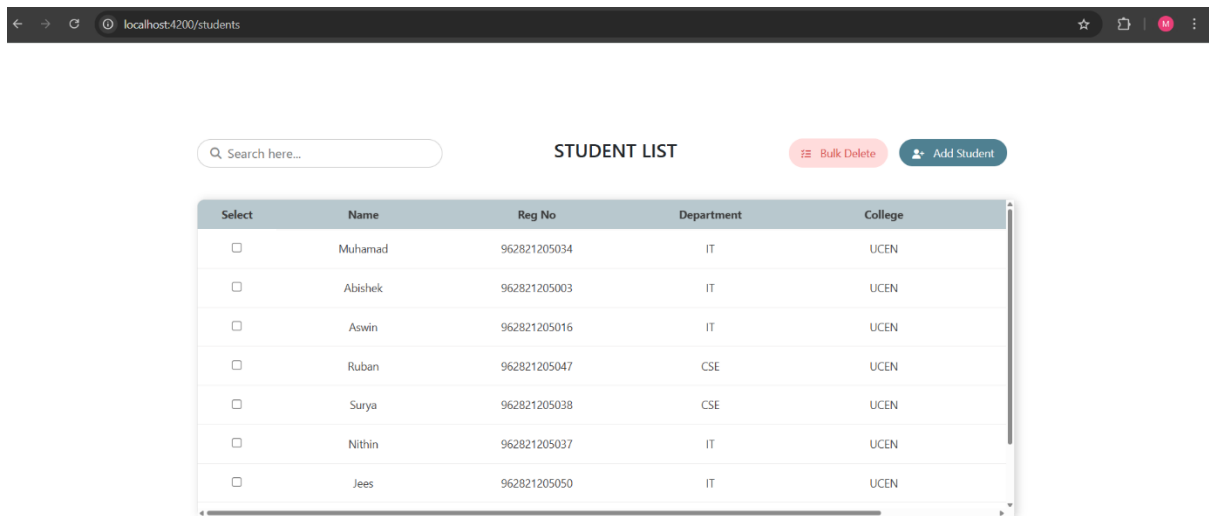
## ember-keyboard

```
<div id="formCont" {{on-key 'Enter' this.handleEnter}} {{on-key 'Escape' @closeForm}}>
```

## liquid-fire

```
<div  
  class="{{if (or (eq this.currentRoute 'students.add') (eq  
this.currentRoute 'students.edit')) 'full-screen-outlet' 'normal-outlet'}}">  
  {{liquid-outlet}}  
</div>
```

## Bidirectional scrollable div and scrollable body:



### index.html

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

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

```
<title>EmberStudentList</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-student-list.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" />
```

```
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
```

```
    {{content-for "head-footer"}}
  </head>

  <body>

    {{content-for "body"}}

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

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

    {{content-for "body-footer"}}
  </body>
</html>
```

### **templates/application.hbs**

```
{{outlet}}
```

### **templates/students.hbs**

```
{{!-- Implemented ember-cli-flash --}}

{{#each this.flashMessages.queue as |flash|}}
  <FlashMessage @flash={{flash}} @messageStyle='bootstrap' />
{{/each}}

{{!-- Implemented ember-truth-helpers --}}

<div

  class="{{if (or (eq this.currentRoute 'students.add') (eq this.currentRoute 'students.edit'))
'full-screen-outlet' 'normal-outlet'}}">

  {{!-- Implemented liquid-fire --}}

  {{liquid-outlet}}

</div>
```

```

<div class="container">
  <div class="search-dlt-cont">
    <div class="search-cont">
      <i class="fa-solid fa-magnifying-glass"></i>
      <input type="text" id="search" placeholder="Search here..." {{on "keyup"
this.handleSearch}}>
    </div>
    <h4>STUDENT LIST</h4>
    <div>
      <button {{on "click" this.bulkDelete}} class="btn-cd dlt">
        <i class="fa-solid fa-list-check"></i> Bulk Delete
      </button>
      <LinkTo @route="students.add" class="btn-cd">
        <i class="fa-solid fa-user-plus"></i> Add Student
      </LinkTo>
    </div>
  </div>
  <StudentList @students={{this.students.filteredStudents}} @deleteRow={{this.deleteRow}}
@editRow={{this.editRow}}
  @updateSelection={{this.updateSelection}} />
</div>

```

### templates/students/add.hbs

```

<div id="overlay"></div>

{{!-- Implemented ember-click-outside--}}

<div {{on-click-outside this.closeForm}}>

```

```
<StudentForm @addRow={{this.addRow}} @closeForm={{this.closeForm}} />
</div>
```

### **templates/students/edit.hbs**

```
<div id="overlay"></div>
```

```
<div {{on-click-outside this.closeForm}}>
```

```
<StudentForm @editStudent={{this.model}} @students={{this.students.students}}
@saveStudent={{this.saveStudent}}
```

```
@updateStudents={{this.updateStudents}} @closeForm={{this.closeForm}} />
</div>
```

### **services/students.js**

```
import Service from '@ember/service';
```

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

```
export default class StudentsService extends Service {
```

```
@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 filteredStudents = this.students;

    addStudent(student) {
      this.students = [...this.students, student];
      this.filteredStudents = this.students;
    }

    searchStudents(query) {
      query = query.toLowerCase();

      this.filteredStudents = this.students.filter(student => {
        const reg = student.reg ? student.reg.toString().toLowerCase() : "";
        return student.name.toLowerCase().includes(query) ||
          reg.includes(query) ||
          student.dept.toLowerCase().includes(query) ||
          student.clg.toLowerCase().includes(query);
      });
    }
  }
}

```

### **routes/students.js**

```

import Route from '@ember/routing/route';
export default class StudentsRoute extends Route {}

```

### **routes/students/add.js**

```

import Route from '@ember/routing/route';
import { inject as service } from '@ember/service';

```



```
export default class StudentsAddRoute extends Route {  
  @service students;  
  model() {  
    return { students:this.students.students, newStudent: { name: "", reg: "", dept: "", clg: "" } };  
  }  
}
```

### **routes/students/edit.js**

```
import Route from '@ember/routing/route';  
import { inject as service } from '@ember/service';  
export default class StudentsEditRoute extends Route {  
  @service students;  
  model(params) {  
    return this.students.students.find(student => student.reg == Number(params.reg));  
  }  
}
```

### **controllers/students.js**

```
import Controller from '@ember/controller';  
import { action } from '@ember/object';  
import { tracked } from '@glimmer/tracking';  
import { service } from '@ember/service';  
import { task, timeout } from 'ember-concurrency';  
  
export default class StudentsController extends Controller {  
  
  @service router;  
  @service students;
```

```
@service flashMessages;

@tracked isFormVisible = false;

@tracked editStudent = null;
```

```
get currentRoute() {
  return this.router.currentRouteName;
}
```

// Implemented ember-concurrency

```
@task

*searchTable(event) {
  let input = event.target.value.toLowerCase();
  yield timeout(300);
  this.students.searchStudents(input);
}
```

```
@action handleSearch(event) {
  this.searchTable.perform(event);
}
```

```
@action bulkDelete() {
  let checkboxes = document.querySelectorAll("#table input[type='checkbox']:checked");
  if (!checkboxes.length) {
    this.flashMessages.warning('Select one or more rows to bulk delete', { timeout: 2000
  });
  }
  return;
}
```

```
checkboxes.forEach(checkbox => {  
  let row = checkbox.parentElement.parentElement;  
  this.deleteRow(row.cells[2].textContent);  
  row.remove();  
});  
}
```

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

```
@action editRow(reg) {  
  this.editStudent = this.students.filteredStudents.find(student => student.reg === reg);  
  this.router.transitionTo('students.edit', reg);  
}  
}
```

### **controllers/students/add.js**

```
import Controller from '@ember/controller';  
import { service } from '@ember/service';  
import { action } from '@ember/object';  
  
export default class StudentsAddController extends Controller {  
  @service router;  
  @service students;  
  @service flashMessages;
```

```
@action addRow(student) {  
  this.students.addStudent(student);  
  this.router.transitionTo('students');  
  //Implemented ember-cli-flash  
  this.flashMessages.success("Student added successfully!");  
}  
  
@action closeForm() {  
  this.router.transitionTo('students');  
}  
}
```

### **controllers/students/edit.js**

```
import Controller from '@ember/controller';  
import { service } from '@ember/service';  
import { action } from '@ember/object';  
  
export default class StudentsEditController extends Controller {  
  @service router;  
  @service students;  
  @service flashMessages;  
  
  @action updateStudents(updatedStudents) {  
    this.students.filteredStudents = updatedStudents;  
    this.router.transitionTo('students');  
    this.flashMessages.success("Student updated successfully!");  
  }  
}
```

```

    }

    @action closeForm() {
      this.router.transitionTo('students');
    }
  }
}

```

### **components/student-form.hbs**

```

{{!-- Implemented ember-keyboard --}}
<div id="formCont" {{on-key 'Enter' this.handleEnter}} {{on-key 'Escape' @closeForm}}>
  <span id="close-btn" {{on "click" @closeForm}}>
    <i class="fa-solid fa-circle-xmark"></i>
  </span>

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

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

  {{!-- Implemented ember-power-select --}}
  <PowerSelect @options={{this.departments}} @selected={{this.dept}}
@onChange={{this.updateDept}}

    @renderInPlace={{true}} @triggerClass="custom-power-select" as
|department|>{{department}}</PowerSelect>

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

```

```
<div class="decision">

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

  <button class="btn-cd" type="button" {{on "click" this.saveStudent}}>

    {{if this.isEditing "Update" "Create"}}

  </button>

</div>

</div>
```

### **components/student-list.hbs**

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

  <table id="table">

    <thead>

      <tr>

        <th>Select</th>

        <th>Name</th>

        <th>Reg No</th>

        <th>Department</th>

        <th>College</th>

        <th>Action</th>

      </tr>

    </thead>

    <tbody>

      {{#if (get @students "length")}}

        {{!-- Implemented @html-next/vertical-collection --}}

        <VerticalCollection @estimateHeight={{50}} @items={{@students}} @key="reg"
```

```

@containerSelector=".table-container" 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 class="btn-ed edit" type="button" {{on "click" (fn @editRow
student.reg)}}>

      <i class="fa-solid fa-pen-to-square"></i>

    </button>

    <button class="btn-ed delete" type="button" {{on "click" (fn @deleteRow
student.reg)}}>

      <i class="fa-solid fa-trash"></i>

    </button>

  </td>

</tr>

</VerticalCollection>

{{else}}

<tr>

  <td colspan="6">No students found</td>

</tr>

{{/if}}

</tbody>

</table>

</div>

```

## **components/student-form.js**

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

export default class StudentFormComponent extends Component {
  @tracked name = this.args.editStudent?.name || '';
  @tracked reg = this.args.editStudent?.reg || '';
  @tracked dept = this.args.editStudent?.dept || 'Select Dept';
  @tracked clg = this.args.editStudent?.clg || '';
  @tracked isEditing = !!this.args.editStudent;
  @service flashMessages;

  departments = ['IT', 'CSE', 'ECE', 'MECH'];

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

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

  @action updateDept(selectedDept) {
    this.dept = selectedDept;
  }
}
```



```
@action updateClg(event) {
```

```
  this.clg = event.target.value;
```

```
}
```

```
@action saveStudent() {
```

```
  if (!this.name || !this.reg || !this.dept || !this.clg) {
```

```
    this.flashMessages.warning("Fill all the details");
```

```
    return;
```

```
}
```

```
if (this.isEditing) {
```

```
  let updatedStudents = this.args.students.map(student =>
```

```
    student.reg === this.reg
```

```
      ? { ...student, name: this.name, dept: this.dept, clg: this.clg }
```

```
      : student
```

```
);
```

```
  this.args.updateStudents(updatedStudents);
```

```
} else {
```

```
  this.args.addRow({
```

```
    name: this.name,
```

```
    reg: this.reg,
```

```
    dept: this.dept,
```

```
    clg: this.clg
```

```
  });
```

```
}
```

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

```
}

@action
handleEnter(event) {
  event.preventDefault();
  if (event.key === "Enter") {
    this.saveStudent();
  }
}

}
```

### **styles/app.css**

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

```
.custom-power-select {
  border: 1px solid #ccc;
  padding: 10px;
  cursor: pointer;
  background-color: white;
  border-radius: 4px;
  font-size: 14px;
  position: relative;
}
```

```
.ember-basic-dropdown-content {  
  background: rgb(247, 247, 247);  
  border: 1px solid #ccc;  
  box-shadow: 0px 4px 6px rgba(0, 0, 0, 0.1);  
  border-radius: 4px;  
  position: absolute;  
  width: 80%;  
  max-height: 200px;  
  overflow-y: auto;  
}
```

```
.ember-power-select-option {  
  padding: 10px;  
  cursor: pointer;  
  font-size: 14px;  
  color: #333;  
}
```

```
.ember-power-select-option[aria-selected="true"] {  
  background-color: #cfcfcf;  
}
```

```
.ember-power-select-option:hover {  
  background-color: #f1f1f1;  
}
```

```
.ember-power-select-options {  
  list-style: none;
```

```
margin: 0;
padding: 0;
}
```

```
.ember-power-select-trigger {
  display: flex;
  align-items: center;
  justify-content: space-between;
}
```

```
body {
  display: flex;
  justify-content: center;
  align-items: center;
  flex-direction: column;
  height: 100vh;
  font-family: Arial, Helvetica, sans-serif;
}
```

```
#overlay {
  position: fixed;
  top: 0;
  left: 0;
  height: 100%;
  width: 100%;
  background-color: rgba(110, 110, 110, 0.2);
  z-index: 1000;
}
```

```
.flash-message {  
  z-index: 1005;  
}
```

```
.container {  
  display: flex;  
  align-items: center;  
  flex-direction: column;  
  padding: 2rem;  
  background-color: #fff;  
  border-radius: 10px;  
  width: 70%;  
  height: 80vh;  
  position: relative;  
}
```

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

```
.normal-outlet {  
  width: 100%;  
  height: auto;  
  position: relative;  
}
```

```
.full-screen-outlet {  
    position: fixed;  
    top: 0;  
    left: 0;  
    width: 100vw;  
    height: 100vh;  
    background-color: rgba(255, 255, 255, 1);  
    z-index: 1000;  
}
```

```
#formCont {  
    display: flex;  
    flex-direction: column;  
    gap: 2rem;  
    top: 15%;  
    left: 35%;  
    background-color: white;  
    padding: 3rem;  
    border-radius: 10px;  
    width: 30%;  
    z-index: 2000;  
    position: relative;  
    margin: 10rem 0;  
}
```

```
.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: #fff;  
  outline: none;  
}
```

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

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

```
.table-container {  
    overflow: auto;  
    width: 80%;  
    border-radius: 10px;  
    box-shadow: 3px 2px 10px #d3d3d3;  
    margin: 1.5rem;  
    background: white;  
    border: 1px solid #e4e4e4;  
    scrollbar-width: thin;  
  
}
```

```
table {  
    margin: 0;  
    background-color: #fff;  
    border-radius: 10px;  
    border-collapse: collapse;  
    color: #3e3e3e;  
    width: 100%;  
    min-width: 1200px;  
    table-layout: fixed;  
  
}
```

```
thead {  
    position: sticky;  
    top: 0;  
    background-color: #b8c8ce;
```



```
    z-index: 5;
    text-align: center;
    font-size: 14px;
}
```

```
tbody {
    overflow-y: auto;
    width: 100%;
    scrollbar-width: thin;
}
```

```
tr th {
    padding: 0.5rem;
}
```

```
table tr {
    border-bottom: 1px solid #eeceded;
}
```

```
td {
    padding: 0.8rem;
    font-size: 14px;
    text-align: center;
    word-wrap: break-word;
}
```

```
th:first-child {
    background-color: #b8c8ce;
```

```
}
```

```
td:first-child {  
    background-color: #ffffff;  
}
```

```
th:first-child, td:first-child {  
    position: sticky;  
    left: 0;  
    top: 0;  
    z-index: 3;  
    width: 100px;  
}
```

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

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

```
}
```

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

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

```
.cancel {  
  background-color: white;  
  color: black;  
  border: 1px solid #9f9f9f;  
}
```

```
.btn-ed {  
  background: transparent;  
  border: none;  
  cursor: pointer;  
}
```

```
.delete {  
  color: #d84949;  
}
```

```
.edit {
```

```
    color: #3e3e3e;  
}
```

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

```
.search-dlt-cont {  
    width: 80%;  
    display: flex;  
    justify-content: space-between;  
    align-items: center;  
    margin: 1rem 0;  
    gap: 1rem;  
}
```

```
#search {  
    padding: 5px 0;  
    border: none;  
    width: 100%;  
}
```

```
.search-cont {  
    display: flex;  
    align-items: center;  
    border: 1px solid #ccc;  
    width: 30%;  
    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": "^7.0.0"
```

```
"ember-source": "~5.9.0",  
"ember-template-imports": "^4.0.0"
```