

OUTPUT:

Table search with selection option

Student List						
<input type="checkbox"/>		Customize Columns		Filters		Search here...
<input type="checkbox"/>		Search by		Delete Selected		Add Student
<input type="checkbox"/>	Reg No	Name	Department			College
<input type="checkbox"/>	962821205001	Aakash	IT			UCEN
<input type="checkbox"/>	962821205002	Abishek	IT			UCEN
<input type="checkbox"/>	962821205003	Ajay	CSE	Spring Boot, MySQL		UCEN
<input type="checkbox"/>	962821205004	Arun	IT	React, Spring Boot		UCEN

Table search by name(specific column):

Student Management System						
<input type="checkbox"/>		Customize Columns		Filters		Search here...
<input type="checkbox"/>		Search by		Delete Selected		Add Student
<input type="checkbox"/>	Reg No	Name	Department			College
<input type="checkbox"/>	962821205031	Muhamad	IT			UCEN

Table search by name and skills(multiple columns):

Student List						
<input type="checkbox"/>		Customize Columns		Filters		Search here...
<input type="checkbox"/>		Search by		Delete Selected		Add Student
<input type="checkbox"/>	Reg No	Name	Department	Skills		College
<input type="checkbox"/>	962821205002	Abishek	IT	Spring Boot, Flutter		UCEN
<input type="checkbox"/>	962821205003	Ajay	CSE	Spring Boot, MySQL		UCEN
<input type="checkbox"/>	962821205004	Arun	IT	React, Spring Boot		UCEN
<input type="checkbox"/>	962821205007	Bhavesh	IT	Flutter, Spring Boot		UCEN
<input type="checkbox"/>	962821205009	Deepak	IT	JavaScript, Spring Boot		UCEN
<input type="checkbox"/>	962821205014	Faiz	IT	Spring Boot, CSS		UCEN
<input type="checkbox"/>	962821205021	Irfan	CSE	Spring Boot, JavaScript		UCEN
<input type="checkbox"/>	962821205022	Jagan	IT	React, Spring Boot		UCEN
<input type="checkbox"/>	962821205027	Kavin	CSE	Spring Boot, React		UCEN

index.html

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

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

```
    <title>Ember-Student-List</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}}  
  
<BasicDropdownWormhole />
```

templates/students.hbs

```
<nav class="navbar">  
  <div class="navbar-title"><i class="fa-solid fa-graduation-cap"></i>Student Management  
System</div>  
  <div class="toggle-profile">  
    <button id="theme-toggle-btn" {{on "click" this.toggleTheme}}>  
      {{#if this.isDarkMode}}  
        <i class="bi bi-sun"></i>  
      {{else}}  
        <i class="bi bi-moon-stars"></i>  
      {{/if}}  
    </button>  
    <i class="bi bi-person-circle"></i>  
  </div>  
</nav>  
  
{{#each this.flashMessages.queue as |flash|}}  
<FlashMessage @flash={{flash}} @messageStyle='bootstrap' />  
{{/each}}  
  
<div
```

```

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

  {{liquid-outlet}}

</div>

<div class="student-container">

  <div class="search-dlt-cont">

    <div><span>Student List</span></div>

    <div class="search-cd-cont">

      <button class="btn-filter" {{on "click" this.toggleCustomize}}>

        <i class="bi bi-layout-sidebar-inset"></i> Customize Columns

      </button>

      {{#if this.isCustomizeVisible}}

      <div class="customize-column" {{on-click-outside this.toggleCustomize}}>

        <span {{on "click" this.toggleCustomize}} {{on-key 'Escape' this.toggleCustomize}}><i

          class="fa-solid fa-xmark"></i></span>

        <div class="filter-section">

          <strong>Columns</strong>

          <div class="custom-column">

            {{#each (array "reg" "name" "dept" "skills" "college") as |col|}}

            <label>

              <input type="checkbox" checked={{this.isColumnVisible col}} {{on "change"

(fn

              this.toggleColumnVisibility col)}} />

              {{capitalize col}}

            </label>

            {{/each}}

          </div>

```

```
</div>
```

```
</div>
```

```
{{/if}}
```

```
<button class="btn-filter" {{on "click" this.toggleFilter}}>
```

```
  <i class="bi bi-funnel"></i> Filters
```

```
</button>
```

```
{{#if this.isFilterVisible}}
```

```
<div class="advanced-filters" {{on-click-outside this.toggleFilter}}>
```

```
  <span {{on "click" this.toggleFilter}} {{on-key 'Escape' this.toggleFilter}}><i
```

```
    class="fa-solid fa-xmark"></i></span>
```

```
<div class="filter-section">
```

```
  <strong>Departments</strong>
```

```
<div class="filter-dept">
```

```
  {{#each this.availableDepartments as |dept|}}
```

```
    <label>
```

```
      <input type="checkbox" value={{dept}} {{on "change" this.updateDeptFilter}}
```

```
        checked={{this.isDeptChecked dept}} />
```

```
        {{dept}}
```

```
    </label>
```

```
  {{/each}}
```

```
</div>
```

```
</div>
```

```
<div class="filter-section">
```

```
  <strong>Skills:</strong>
```

```
<div class="filter-skill">
```

```

        {{#each this.availableSkills as |skill|}}

        <label>

            <input type="checkbox" value={{skill}} {{on "change" this.updateSkillsFilter}}

                checked={{includes this.filter.skills skill}} />

            {{skill}}

        </label>

        {{/each}}

    </div>

</div>

    <button type="button" {{on "click" this.applyFilter}} class="btn-cd filter">Apply
Filters</button>

    <button type="button" {{on "click" this.clearAllFilters}} class="btn-cd cancel">Clear
All</button>

</div>

{{/if}}

<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 class="power-select-wrapper">

        <PowerSelectMultiple @selected={{this.selectedColumnValues}}

            @options={{this.searchColumnOptions}}

            @onChange={{this.updateSearchColumns}}

            @placeholder="Search by" @searchEnabled={{false}} as |option| >

                {{option.label}}

            </PowerSelectMultiple>

        </div>

    </div>

```

```

        <button {{on "click" this.bulkDelete}} class="btn-cd dlt" {{on-key 'Delete'
this.bulkDelete}}>
            <i class="fa-regular fa-trash-can"></i> Delete Selected
        </button>

        <LinkTo @route="students.add" class="btn-cd add" {{on-key 'alt+a' this.addStud}}>
            <i class="fa-solid fa-user-plus"></i> Add Student
        </LinkTo>
    </div>
</div>

<StudentList @students={{this.students.filteredStudents}} @deleteRow={{this.deleteRow}}
@editRow={{this.editRow}}

    @toggleSelectAll={{this.toggleSelectAll}} @selectAll={{this.selectAll}}
@sortByName={{this.sortByName}}

    @isAscending={{this.isAscending}} @visibleColumns={{this.visibleColumns}} />
</div>

```

templates/students/add.hbs

```

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

```

templates/students/edit.hbs

```

<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: "Aakash", reg: 962821205001, dept: "IT", clg: "UCEN", skills: ["JavaScript",  
"React"] },
```

```
    { name: "Abishek", reg: 962821205002, dept: "IT", clg: "UCEN", skills: ["Spring Boot",  
"Flutter"] },
```

```
    { name: "Ajay", reg: 962821205003, dept: "CSE", clg: "UCEN", skills: ["Spring Boot",  
"MySQL"] },
```

```
    { name: "Arun", reg: 962821205004, dept: "IT", clg: "UCEN", skills: ["React", "Spring  
Boot"] },
```

```
    { name: "Aswin", reg: 962821205005, dept: "IT", clg: "UCEN", skills: ["Node.js",  
"MySQL"] },
```

```
    { name: "Bala", reg: 962821205006, dept: "CSE", clg: "UCEN", skills: ["CSS", "Node.js"] },
```

```
    { name: "Bhavesh", reg: 962821205007, dept: "IT", clg: "UCEN", skills: ["Flutter", "Spring  
Boot"] },
```

```
    { name: "Charan", reg: 962821205008, dept: "CSE", clg: "UCEN", skills: ["CSS", "MySQL"]  
  },
```

```
    { name: "Deepak", reg: 962821205009, dept: "IT", clg: "UCEN", skills: ["JavaScript",  
"Spring Boot"] },
```

```
    { name: "Dev", reg: 962821205010, dept: "IT", clg: "UCEN", skills: ["React", "HTML"] },
```

```
    { name: "Dinesh", reg: 962821205011, dept: "IT", clg: "UCEN", skills: ["JavaScript",  
"HTML"] },
```

```
    { name: "Ethan", reg: 962821205012, dept: "CSE", clg: "UCEN", skills: ["Node.js",  
"JavaScript"] },
```

```
    { name: "Elavarasan", reg: 962821205013, dept: "IT", clg: "UCEN", skills: ["HTML",  
"React"] },
```

```
    { name: "Faiz", reg: 962821205014, dept: "IT", clg: "UCEN", skills: ["Spring Boot", "CSS"]  
  },
```



```
{ name: "Gokul", reg: 962821205015, dept: "CSE", clg: "UCEN", skills: ["MySQL",  
"Flutter"] },  
  
{ name: "Guhan", reg: 962821205016, dept: "CSE", clg: "UCEN", skills: ["Flutter",  
"React"] },  
  
{ name: "Hari", reg: 962821205017, dept: "CSE", clg: "UCEN", skills: ["CSS", "HTML"] },  
  
{ name: "Harish", reg: 962821205018, dept: "IT", clg: "UCEN", skills: ["Node.js", "CSS"] },  
  
{ name: "Harsha", reg: 962821205019, dept: "IT", clg: "UCEN", skills: ["MySQL", "HTML"]  
},  
  
{ name: "Imran", reg: 962821205020, dept: "CSE", clg: "UCEN", skills: ["Node.js",  
"JavaScript"] },  
  
{ name: "Irfan", reg: 962821205021, dept: "CSE", clg: "UCEN", skills: ["Spring Boot",  
"JavaScript"] },  
  
{ name: "Jagan", reg: 962821205022, dept: "IT", clg: "UCEN", skills: ["React", "Spring  
Boot"] },  
  
{ name: "Jeeva", reg: 962821205023, dept: "IT", clg: "UCEN", skills: ["HTML", "Node.js"]  
},  
  
{ name: "Karthik", reg: 962821205024, dept: "IT", clg: "UCEN", skills: ["Flutter", "React"]  
},  
  
{ name: "Keerthi", reg: 962821205025, dept: "IT", clg: "UCEN", skills: ["React", "CSS"] },  
  
{ name: "Kiran", reg: 962821205026, dept: "IT", clg: "UCEN", skills: ["CSS", "Flutter"] },  
  
{ name: "Kavin", reg: 962821205027, dept: "CSE", clg: "UCEN", skills: ["Spring Boot",  
"React"] },  
  
{ name: "Lalith", reg: 962821205028, dept: "CSE", clg: "UCEN", skills: ["HTML",  
"MySQL"] },  
  
{ name: "Lokesh", reg: 962821205029, dept: "CSE", clg: "UCEN", skills: ["Flutter",  
"MySQL"] },  
  
{ name: "Manoj", reg: 962821205030, dept: "IT", clg: "UCEN", skills: ["JavaScript",  
"React"] },  
  
{ name: "Muhamad", reg: 962821205031, dept: "IT", clg: "UCEN", skills: ["JavaScript",  
"React"] },  
  
{ name: "Naveen", reg: 962821205032, dept: "CSE", clg: "UCEN", skills: ["Spring Boot",  
"Node.js"] },
```

```
{ name: "Nithin", reg: 962821205033, dept: "IT", clg: "UCEN", skills: ["Node.js", "CSS"] },
{ name: "Omprakash", reg: 962821205034, dept: "IT", clg: "UCEN", skills: ["CSS",
"HTML"] },
{ name: "Pranav", reg: 962821205035, dept: "CSE", clg: "UCEN", skills: ["MySQL",
"JavaScript"] },
{ name: "Praveen", reg: 962821205036, dept: "CSE", clg: "UCEN", skills: ["MySQL",
"HTML"] },
{ name: "Raj", reg: 962821205037, dept: "IT", clg: "UCEN", skills: ["React", "Flutter"] },
{ name: "Ragul", reg: 962821205038, dept: "IT", clg: "UCEN", skills: ["Node.js", "Flutter"]
},
{ name: "Ruban", reg: 962821205039, dept: "CSE", clg: "UCEN", skills: ["HTML", "CSS"] },
{ name: "Sanjay", reg: 962821205040, dept: "IT", clg: "UCEN", skills: ["Node.js", "Spring
Boot"] },
{ name: "Sathish", reg: 962821205041, dept: "IT", clg: "UCEN", skills: ["JavaScript",
"Node.js"] },
{ name: "Suriya", reg: 962821205042, dept: "IT", clg: "UCEN", skills: ["HTML",
"JavaScript"] },
{ name: "Surya", reg: 962821205043, dept: "CSE", clg: "UCEN", skills: ["React", "Flutter"]
},
{ name: "Tharun", reg: 962821205044, dept: "CSE", clg: "UCEN", skills: ["HTML", "CSS"]
},
{ name: "Udhay", reg: 962821205045, dept: "IT", clg: "UCEN", skills: ["JavaScript",
"MySQL"] },
{ name: "Venkatesh", reg: 962821205046, dept: "CSE", clg: "UCEN", skills: ["React",
"Flutter"] },
{ name: "Vijay", reg: 962821205047, dept: "IT", clg: "UCEN", skills: ["Flutter", "MySQL"]
},
{ name: "Vimal", reg: 962821205048, dept: "CSE", clg: "UCEN", skills: ["React", "CSS"] },
{ name: "Yogesh", reg: 962821205049, dept: "IT", clg: "UCEN", skills: ["Spring Boot",
"CSS"] },
{ name: "Zaid", reg: 962821205050, dept: "CSE", clg: "UCEN", skills: ["Node.js", "HTML"]
},
```

```
];
```

```
@tracked filteredStudents = this.students;
```

```
searchStudents(query, searchColumns = ['name', 'dept', 'clg']) {
```

```
    query = query.toLowerCase();
```

```
    this.filteredStudents = this.students.filter(student => {
```

```
        return searchColumns.some(column => {
```

```
            switch(column) {
```

```
                case 'reg':
```

```
                    return student.reg.toString().toLowerCase().includes(query);
```

```
                case 'skills':
```

```
                    return student.skills.join(' ').toLowerCase().includes(query);
```

```
                case 'name':
```

```
                case 'dept':
```

```
                case 'clg':
```

```
                    return student[column]?.toLowerCase().includes(query);
```

```
                default:
```

```
                    return false;
```

```
            }
```

```
        });
```

```
    });
```

```
}
```

```
}
```

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';  
  
export default class StudentsAddRoute extends Route { }
```

routes/students/edit.js

```
import Route from '@ember/routing/route';  
  
import { 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';  
  
import { htmlSafe } from '@ember/template';  
  
  
export default class StudentsController extends Controller {
```

```
@service router;
@service students;
@service flashMessages;
@tracked isFormVisible = false;
@tracked editStudent = null;
@tracked selectAll = false;
@tracked isDarkMode = false;
@tracked isAscending = true;
@tracked recentlyDeletedStudents = [];

@tracked isFilterVisible = false;
@tracked filter = {
  depts: [],
  skills: []
};

@tracked isCustomizeVisible = false;
@tracked visibleColumns = ['reg', 'name', 'dept', 'skills', 'college'];

@tracked searchColumns = ['name'];
@tracked lastSearchQuery = "";

searchColumnOptions = [
  { value: 'name', label: 'Name' },
  { value: 'reg', label: 'Reg No' },
  { value: 'dept', label: 'Dept' },
  { value: 'skills', label: 'Skills' },
  { value: 'clg', label: 'College' }
```

```
];
```

```
get isDeptChecked() {  
    return (dept) => {  
        return this.filter.depts.includes(dept);  
    };  
}
```

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

```
@action toggleFilter() {  
    this.isFilterVisible = !this.isFilterVisible;  
}
```

```
@action toggleCustomize() {  
    this.isCustomizeVisible = !this.isCustomizeVisible;  
}
```

```
get availableDepartments() {  
    let depts = this.students.students.map(student => student.dept);  
    return [...new Set(depts)];  
}
```

```
get availableSkills() {  
    let allSkills = this.students.students.flatMap(student => student.skills || []);  
    return [...new Set(allSkills)];  
}
```

```
}
```

```
@action applyFilter() {
```

```
  this.isFilterVisible = !this.isFilterVisible;
```

```
  this.students.filteredStudents = this.students.students.filter(student => {
```

```
    let { depts, skills } = this.filter;
```

```
    if (depts.length && !depts.includes(student.dept)) {
```

```
      return false;
```

```
    }
```

```
    if (skills.length && !skills.every(skill => student.skills.includes(skill))) {
```

```
      return false;
```

```
    }
```

```
    return true;
```

```
  });
```

```
  this.flashMessages.success(htmlSafe('<i class="bi bi-check-circle-fill"></i> Filters applied'));
}
```

```
@action clearAllFilters() {
```

```
  this.filter = {
```

```
    depts: [],
```

```
    skills: []
```

```
  };
```

```
  this.isFilterVisible = false;
```

```
  this.students.filteredStudents = [...this.students.students];
```

```
}
```

```
@action updateDeptFilter(event) {  
  let value = event.target.value;  
  let isChecked = event.target.checked;  
  
  if (isChecked) {  
    this.filter = {  
      ...this.filter,  
      depts: [...this.filter.depts, value]  
    };  
  } else {  
    this.filter = {  
      ...this.filter,  
      depts: this.filter.depts.filter(dept => dept !== value)  
    };  
  }  
}
```

```
@action updateSkillsFilter(event) {  
  let value = event.target.value;  
  let isChecked = event.target.checked;  
  
  if (isChecked) {  
    this.filter = {  
      ...this.filter,  
      skills: [...this.filter.skills, value]  
    };  
  } else {
```



```
this.filter = {  
  ...this.filter,  
  skills: this.filter.skills.filter(skill => skill !== value)  
};  
}  
}
```

```
@action toggleColumnVisibility(column) {  
  if (this.visibleColumns.includes(column)) {  
    this.visibleColumns = this.visibleColumns.filter(c => c !== column);  
  } else {  
    this.visibleColumns = [...this.visibleColumns, column];  
  }  
}
```

```
@action isColumnVisible(column) {  
  return this.visibleColumns.includes(column);  
}
```

```
@action sortByName() {  
  const sorted = [...this.students.filteredStudents].sort((a, b) => {  
    return this.isAscending  
      ? b.name.localeCompare(a.name)  
      : a.name.localeCompare(b.name);  
  });  
  console.log(this.isAscending)  
  
  this.students.filteredStudents = sorted;
```

```
    this.isAscending = !this.isAscending;
  }
```

```
@action addStud() {
  this.router.transitionTo('students.add');
}
```

```
@action toggleSelectAll(event) {
  this.selectAll = event.target.checked;
```

```
  let checkboxes = document.querySelectorAll("#table input[type='checkbox']");
  checkboxes.forEach(checkbox => {
    checkbox.checked = this.selectAll;
  });
}
```

```
@action handleSearch(event) {
  const query = event.target.value.toLowerCase();
  this.lastSearchQuery = query;
  this.students.searchStudents(query, this.searchColumns);
}
```

```
get selectedColumnValues() {
  return this.searchColumns.map(value => {
    return this.searchColumnOptions.find(opt => opt.value === value);
  }).filter(Boolean);
}
```

```

@action
updateSearchColumns(selected) {
  this.searchColumns = selected.map(item => item.value);

  if (this.lastSearchQuery) {
    this.students.searchStudents(this.lastSearchQuery, this.searchColumns);
  }
}

@action bulkDelete() {
  let checkboxes = document.querySelectorAll("#table input[type='checkbox']:checked");

  if (!checkboxes.length) {
    this.flashMessages.warning(htmlSafe('<i class="bi bi-exclamation-triangle-fill"></i> Select
one or more rows to delete'));
    return;
  }

  let selectedRegs = Array.from(checkboxes).map(checkbox => {
    let row = checkbox.closest("tr");
    return row.cells[1].textContent.trim();
  });

  this.students.filteredStudents = this.students.filteredStudents.filter(
    student => !selectedRegs.includes(String(student.reg)));

  this.selectAll = false;
  checkboxes.forEach(checkbox => checkbox.checked = false);

```

```
    this.flashMessages.success(htmlSafe('<i class="bi bi-check-circle-fill"></i> Selected  
students deleted successfully', { timeout: 2000 }));
```

```
}
```

```
@action deleteRow(reg) {
```

```
    const index = this.students.filteredStudents.findIndex(s => s.reg === reg);
```

```
    const student = this.students.filteredStudents[index];
```

```
    this.students.filteredStudents = this.students.filteredStudents.filter(s => s.reg !== reg);
```

```
    const id = Date.now();
```

```
    this.recentlyDeletedStudents.push({ id, student, index });
```

```
    this.flashMessages.add({
```

```
        type: 'success',
```

```
        message: htmlSafe(`<i class="bi bi-check-circle-fill"></i> Student deleted. <button  
class="undo-btn" data-id="${id}">Undo</button>`),
```

```
        timeout: 5000,
```

```
        sticky: true,
```

```
    });
```

```
    setTimeout(() => {
```

```
        const undoBtn = document.querySelector(`.undo-btn[data-id="${id}"]`);
```

```
        if (undoBtn) {
```

```
            undoBtn.addEventListener('click', () => this.undoDelete(id));
```

```
        }
```

```
    }, 0);
```

```
setTimeout(() => {  
  this.recentlyDeletedStudents = this.recentlyDeletedStudents.filter(item => item.id !== id);  
}, 5000);  
}
```

```
@action undoDelete(id) {  
  const entry = this.recentlyDeletedStudents.find(item => item.id === id);  
  if (entry) {  
    const studentsCopy = [...this.students.filteredStudents];  
    studentsCopy.splice(entry.index, 0, entry.student);  
    this.students.filteredStudents = studentsCopy;  
  
    this.recentlyDeletedStudents = this.recentlyDeletedStudents.filter(  
      item => item.id !== id  
    );  
  
    this.flashMessages.success(htmlSafe('<i class="bi bi-check-circle-fill"></i> Undo  
successful!'));  
  }  
}
```

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

```
@action toggleTheme() {  
  this.isDarkMode = !this.isDarkMode;
```

```
const newTheme = this.isDarkMode ? 'dark' : 'light';

document.documentElement.setAttribute('data-theme', newTheme);

}

}
```

controllers/students/add.js

```
import Controller from '@ember/controller';
import { service } from '@ember/service';
import { action } from '@ember/object';
import { htmlSafe } from '@ember/template';

export default class StudentsAddController extends Controller {

  @service router;

  @service students;

  @service flashMessages;

  @action addRow(student) {

    this.students.students = [...this.students.students, student];

    this.students.filteredStudents = this.students.students;

    this.router.transitionTo('students');

    this.flashMessages.success(htmlSafe('<i class="bi bi-check-circle-fill"></i> 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';
import { htmlSafe } from '@ember/template';

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(htmlSafe('<i class="bi bi-check-circle-fill"></i> Student
updated successfully!'));
  }

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

components/student-form.hbs

```
<form id="formCont" {{on "keydown" this.handleEnter}} {{on-key 'Escape' @closeForm}}
{{on-click-outside @closeForm}}>
```

```
<h4>{{if this.isEditing "Edit Student" "Add New Student"}}</h4>
```

```
<span id="close-btn" {{on "click" @closeForm}}>
```

```
<i class="fa-solid fa-xmark"></i>
```

```
</span>
```

```
<label for="name">Name</label>
```

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

```
<label for="reg">Registration Number</label>
```

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

```
<label for="clg">College</label>
```

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

```
<label for="dept">Department</label>
```

```
<PowerSelect @options={{this.departments}} @selected={{this.dept}}
@searchEnabled={{true}}
@onChange={{this.updateDept}} @placeholder="Select dept..." @triggerId="dept"
@required={{true}} as |department|>
```

```
  {{department}}
```

```
</PowerSelect>
```

```
<label for="skills">Skills</label>
```



```

<PowerSelectMultiple @options={{this.skills}} @selected={{this.selectedSkills}}
@searchEnabled={{true}}

    @onChange={{this.updateSkills}} @placeholder="Select skills..." @triggerId="skills"
@required={{true}} as |skill|>

    {{skill}}

</PowerSelectMultiple>

<div class="decision">

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

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

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

    </button>

</div>

</form>

```

components/student-list.hbs

```

<div class="table-container">

    <table id="table">

        <thead>

            <tr>

                <th><input type="checkbox" {{on "change" @toggleSelectAll}} checked={{@selectAll}}
/></th>

                {{#if (includes @visibleColumns "reg")}}<th>Reg No</th>{{/if}}

                {{#if (includes @visibleColumns "name")}}<th>Name <i class={{if @isAscending "bi bi-
sort-alpha-down" "bi bi-sort-alpha-down-alt" }} {{on "click" @sortByName}}></i></th>{{/if}}

                {{#if (includes @visibleColumns "dept")}}<th>Department</th>{{/if}}

                {{#if (includes @visibleColumns "skills")}}<th>Skills</th>{{/if}}

                {{#if (includes @visibleColumns "college")}}<th>College</th>{{/if}}

```

```

        <th>Action</th>

    </tr>

</thead>

<tbody>

    {{#if (and @students (gt @students.length 0))}}

        <VerticalCollection @estimateHeight={{60}} @items={{@students}} @key="reg"
@containerSelector=".table-container"

        as |student|>

            <tr>

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

                {{#if (includes @visibleColumns "reg")}}<td>{{student.reg}}</td>{{/if}}

                {{#if (includes @visibleColumns "name")}}<td>{{student.name}}</td>{{/if}}

                {{#if (includes @visibleColumns "dept")}}<td>{{student.dept}}</td>{{/if}}

                {{#if (includes @visibleColumns "skills")}}<td>{{join-array student.skills " "}}</td>{{/if}}

                {{#if (includes @visibleColumns "college")}}<td>{{student.clg}}</td>{{/if}}

                <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" style="text-align: center;">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';
import { htmlSafe } from '@ember/template';

export default class StudentFormComponent extends Component {
  @service flashMessages;

  @tracked name = this.args.editStudent?.name || '';
  @tracked reg = this.args.editStudent?.reg || '';
  @tracked selectedSkills = this.args.editStudent?.skills || [];
  @tracked dept = this.args.editStudent?.dept || null;
  @tracked clg = this.args.editStudent?.clg || '';
  @tracked isEditing = !!this.args.editStudent;

  skills = ['JavaScript', 'React', 'Spring Boot', 'Flutter', 'MySQL', 'Node.js', 'CSS', 'HTML'];

  @action updateSkills(selected) {
```

```
    this.selectedSkills = selected;
    console.log(this.selectedSkills);
}
```

```
departments = ['IT', 'CSE', 'ECE', 'MECH'];
@action updateDept(selectedDept) {
    this.dept = selectedDept;
}
```

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

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

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

```
@action saveStudent() {

    if (!this.name || !this.reg || !this.dept || !this.clg || this.selectedSkills.length === 0) {
        this.flashMessages.warning(htmlSafe('<i class="bi bi-exclamation-triangle-fill"></i> 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, skills: this.selectedSkills, clg:
this.clg }
            : student
        );

        this.args.updateStudents(updatedStudents);
    } else {
        this.args.addRow({
            name: this.name,
            reg: this.reg,
            dept: this.dept,
            skills: this.selectedSkills,
            clg: this.clg
        });
    }

    this.args.closeForm();
}

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

```

```
}  
  
}
```

styles/app.css

```
:root {  
  --bg-color: #ffffff;  
  --text-color: #353535;  
  --card-bg: #ffffff;  
  --input-bg: #ffffff;  
  --table-bg: #ffffff;  
  --table-header-bg: #f8f9fa;  
  --table-text: #3e3e3e;  
  --border-color: #dadada;  
  --shadow-color: rgba(0, 0, 0, 0.1);  
  --navbar-bg: #ffffff;  
  --hover-bg: #f5f5f5;  
  --search-border: #ccc;  
}  
  
[data-theme='dark'] {  
  --bg-color: #212121;  
  --text-color: #f1f1f1;
```

```
--card-bg: #2b2b2b;
--input-bg: #333333;
--table-bg: #272727;
--table-header-bg: #2b2b2b;
--table-text: #e0e0e0;
--border-color: #3b3b3b;
--shadow-color: rgba(0, 0, 0, 0.6);
--navbar-bg: #2e2e2e;
--hover-bg: #262626;
--search-border: #555555;
}

.ember-power-select-dropdown {
  background-color: var(--card-bg) !important;
  color: var(--text-color) !important;
  border: 1px solid var(--border-color) !important;
  box-shadow: 0 4px 8px var(--shadow-color) !important;
}

.ember-power-select-option {
  color: var(--text-color) !important;
}

.ember-power-select-option[aria-selected='true'] {
  background-color: #4f8091 !important;
  color: white !important;
  font-weight: bold !important;
}
```

```
.ember-power-select-option[aria-current='true'] {  
  background-color: var(--hover-bg) !important;  
  color: var(--text-color) !important;  
}
```

```
.ember-power-select-option:hover {  
  background-color: var(--hover-bg) !important;  
  color: var(--text-color) !important;  
}
```

```
form .ember-power-select-trigger {  
  background-color: var(--card-bg) !important;  
  color: var(--text-color) !important;  
  border: 1px solid var(--border-color) !important;  
  border-radius: 5px;  
  padding: 0 1rem;  
  scrollbar-width: none;  
  height: 40px;  
}
```

```
.ember-power-select-placeholder {  
  color: var(--text-color) !important;  
  opacity: 0.6;  
}
```

```
.ember-power-select-multiple-option {  
  background-color: var(--hover-bg) !important;
```



```
color: var(--text-color) !important;
border: 1px solid var(--border-color) !important;
margin: 0 4px 4px 0;
padding: 2px 6px;
border-radius: 4px;
}
```

```
.ember-power-select-clear-btn {
  color: var(--text-color) !important;
}
```

```
.ember-basic-dropdown-content {
  background-color: var(--card-bg) !important;
  color: var(--text-color) !important;
  border: 1px solid var(--border-color) !important;
}
```

```
body {
  background-color: var(--bg-color) !important;
  color: var(--text-color);
  transition: all 0.3s ease;
  display: flex;
  align-items: center;
  flex-direction: column;
  font-family: Arial, Helvetica, sans-serif;
  height: 100vh;
}
```

```
.navbar {  
  margin-bottom: 2rem;  
  width: 100%;  
  height: 4rem;  
  background-color: var(--navbar-bg);  
  box-shadow: 0 2px 8px var(--shadow-color);  
  font-weight: bold;  
  font-size: 20px;  
  display: flex;  
  justify-content: space-between;  
  color: var(--text-color);  
}
```

```
.navbar-title {  
  display: flex;  
  align-items: center;  
  gap: 5px;  
  padding: 0 2rem;  
  color: var(--text-color);  
  font-size: 18px;  
  font-weight: 600;  
}
```

```
#theme-toggle-btn {  
  background: transparent;  
  border: none;  
  font-size: 14px;  
  font-weight: 600;
```

```
    display: flex;
    justify-content: center;
    align-items: center;
}
```

```
#theme-toggle-btn i {
    font-size: 20px;
}
```

```
.toggle-profile {
    display: flex;
    gap: 1.5rem;
    padding: 0 2rem;
}
```

```
.toggle-profile i {
    font-size: 24px;
    cursor: pointer;
    color: var(--table-text);
}
```

```
.flash-message {
    text-align: center;
    min-width: 250px !important;
    border-radius: 10px !important;
    padding: 0.5rem !important;
    z-index: 1050;
```

```
position: fixed !important;
top: 1rem;
right: 2rem;
opacity: 1;
animation: bounceIn 0.6s ease-out, fadeOut 0.5s ease-in 5s forwards;
}
```

```
@keyframes bounceIn {
  0% {
    transform: scale(0.5);
    opacity: 0;
  }
```

```
  60% {
    transform: scale(1.1);
    opacity: 1;
  }
```

```
  100% {
    transform: scale(1);
    opacity: 1;
  }
}
```

```
@keyframes fadeOut {
  to {
    opacity: 0;
    transform: translateY(-20px);
```

```
}  
}
```

```
.undo-btn {  
  background: transparent;  
  border: none;  
  border-bottom: 1px solid #000;  
  color: rgb(86, 86, 86);  
}
```

```
.student-container {  
  display: flex;  
  align-items: center;  
  flex-direction: column;  
  border: 1px solid var(--border-color);  
  border-radius: 10px;  
  height: 80vh;  
  width: 80%;  
  background-color: var(--card-bg);  
}
```

```
.full-screen-outlet {  
  position: fixed;  
  top: 0;  
  left: 0;  
  height: 100%;  
  width: 100%;  
  background: linear-gradient(to bottom, rgba(50, 50, 50, 0.7), rgba(45, 45, 45, 0.6));  
}
```

```
    z-index: 1000;
}
```

```
[data-theme='dark'] .full-screen-outlet {
    background: linear-gradient(to bottom, rgba(67, 67, 67, 0.6), rgba(88, 88, 88, 0.6));
}
```

```
#formCont {
    display: flex;
    flex-direction: column;
    gap: 0.5rem;
    left: 35%;
    background-color: var(--card-bg);
    color: var(--text-color);
    padding: 2.5rem;
    border-radius: 20px;
    width: 30%;
    z-index: 2000;
    position: relative;
    margin: 4% 0;
}
```

```
label {
    font-weight: 600;
    color: var(--text-color);
    margin: 0.3rem 0;
    font-size: 14px;
}
```

```
input {  
  padding: 0.5rem 1rem;  
  border: 1px solid var(--border-color);  
  background-color: var(--card-bg);  
  color: var(--text-color);  
  outline: none;  
  height: 35px;  
}
```

```
input:focus {  
  border: 1px solid #a6a6a6;  
}
```

```
input::placeholder {  
  color: var(--text-color);  
  opacity: 0.6;  
}
```

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

```
.table-container {  
  overflow: auto;  
  scroll-behavior: smooth;  
  scrollbar-color: var(--border-color) transparent;
```

```
width: 100%;  
box-shadow: 3px 2px 10px var(--shadow-color);  
background: var(--card-bg);  
border: 1px solid var(--border-color);  
scrollbar-width: thin;  
  
}
```

```
.table-container tr {  
    height: auto;  
    min-height: 40px;  
}
```

```
table {  
    background-color: var(--table-bg);  
    border-radius: 10px;  
    border-collapse: collapse;  
    color: var(--table-text);  
    width: 100%;  
    min-width: 1400px;  
    table-layout: fixed;  
}
```

```
thead {  
    position: sticky;  
    top: 0;  
    background-color: var(--table-header-bg);  
    color: var(--text-color);
```



```
text-align: center;
font-size: 14px;
z-index: 10;
}
```

```
tbody {
  overflow-y: auto;
  width: 100%;
  scrollbar-width: thin;
  background-color: var(--table-bg);
  color: var(--table-text);
  font-weight: 600;
}
```

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

```
tr th i {
  cursor: pointer;
  margin-left: 0.5rem;
  font-size: 18px;
}
```

```
table tr td {
  border-bottom: 1px solid var(--border-color);
  padding: 0.8rem;
  font-size: 14px;
```

```
text-align: center;
word-wrap: break-word;
}
```

```
tr:hover {
  background-color: var(--hover-bg);
}
```

```
th:first-child,
td:first-child {
  background-color: var(--table-header-bg);
  position: sticky;
  left: 0;
  top: 0;
  z-index: 1;
  width: 100px;
}
```

```
th:first-child::after,
td:first-child::after {
  content: "";
  position: absolute;
  top: 0;
  right: -5px;
  width: 5px;
  height: 100%;
  background: linear-gradient(to right, rgba(0, 0, 0, 0.05), transparent);
  pointer-events: none;
}
```

```
}
```

```
.btn-cd {  
  padding: 0.4rem 0.8rem;  
  border-radius: 5px;  
  border: none;  
  background-color: #4f8091;  
  color: white;  
  cursor: pointer;  
  font-size: 14px;  
  font-weight: 600;  
  text-decoration: none;  
}
```

```
.btn-cd i {  
  font-size: 12px;  
  margin-right: 5px;  
}
```

```
.cancel,  
.uc {  
  margin: 1rem 0;  
}
```

```
.add {  
  background: linear-gradient(135deg, #4f8091, #2b5a68);  
}
```

```
.dlt {  
  background: linear-gradient(135deg, #e37272, #c94b4b);  
}
```

```
.cancel {  
  background-color: var(--card-bg);  
  color: var(--text-color);  
  border: 1px solid #c8c8c8;  
}
```

```
.btn-ed {  
  background: transparent;  
  border: none;  
  cursor: pointer;  
  margin: 0 0.3rem;  
}
```

```
.btn-ed i {  
  font-size: 14px;  
}
```

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

```
.edit {  
  color: var(--text-color);  
}
```

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

```
.decision {  
    display: flex;  
    width: 100%;  
    gap: 1rem;  
}
```

```
.decision button {  
    width: 100%;  
}
```

```
#close-btn {  
    cursor: pointer;  
    position: absolute;  
    top: 2rem;  
    right: 2rem;  
    color: var(--text-color);  
    font-size: 20px;  
}
```

```
.btn-filter {  
    border: 1px solid var(--search-border);  
    background: transparent;  
    color: var(--text-color);
```

```
    opacity: 0.8;
    border-radius: 5px;
    padding: 0.3rem 1rem;
    font-size: 14px;
    font-weight: 600;
}
```

```
.btn-filter i {
    font-size: 14px;
}
```

```
.advanced-filters button {
    float: right;
}
```

```
.filter {
    margin: 1rem;
}
```

```
.btn-cd:hover,
.btn-filter:hover {
    scale: 1.03;
}
```

```
.advanced-filters,
.customize-column {
    position: absolute;
    top: 100%;
```

```
    left: 12rem;
    z-index: 1000;
    padding: 1rem;
    background: var(--table-header-bg);
    color: var(--text-color);
    border: 1px solid #ccc;
    border-radius: 8px;
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.4);
    margin-top: 0.4rem;
}
```

```
.close-modal {
    position: absolute;
    top: 0.5rem;
    right: 1rem;
    cursor: pointer;
}
```

```
.filter-section {
    display: flex;
    flex-direction: column;
    gap: 0.5rem;
}
```

```
.filter-dept,
.filter-skill,
.custom-column {
    display: grid;
```

```
    grid-template-columns: 1fr 1fr 1fr;
    gap: 1rem;
    margin: 0.5rem;
}
```

```
.title-cont {
    display: flex;
    justify-content: center;
    align-items: center;
}
```

```
.search-dlt-cont {
    width: 98%;
    display: flex;
    justify-content: space-between;
    align-items: center;
    margin: 1rem;
}
```

```
.search-dlt-cont span {
    font-size: 20px;
    font-weight: 700;
    color: var(--text-color);
}
```

```
.search-cont {
    display: flex;
    align-items: center;
```



```
border: 1px solid var(--search-border);  
border-radius: 5px;  
width: 40%;  
background-color: var(--input-bg);  
padding: 0 1rem;  
}
```

```
.search-cont i {  
  font-size: 14px;  
  margin-right: 8px;  
  color: var(--text-color);  
  opacity: 0.7;  
}
```

```
#search {  
  flex: 1;  
  border: none;  
  background: transparent;  
  color: var(--text-color);  
  padding: 0.3rem;  
  min-width: 120px;  
}
```

```
.search-cont .ember-power-select-trigger {  
  background: transparent !important;  
  border: none !important;  
  height: 40px !important;  
  overflow-y: scroll !important;
```

```
    scrollbar-width: none !important;
    min-width: 100px !important;
}
```

```
.search-cont .ember-power-select-multiple-option {
    background-color: var(--hover-bg) !important;
    color: var(--text-color) !important;
    border: none !important;
    margin: 0.5rem 0.2rem !important;
    padding: 0 8px !important;
    font-size: 10px !important;
    border-radius: 20px !important;
    font-weight: 500 !important;
    display: inline-flex;
    align-items: center;
    gap: 4px;
    box-shadow: 0 1px 3px rgba(0, 0, 0, 0.1);
}
```

```
.search-cont .ember-power-select-placeholder {
    font-size: 12px !important;
    font-weight: 400 !important;
    opacity: 0.7 !important;
    color: var(--text-color) !important;
    margin: 0.5rem 0 !important;
}
```

```
.ember-power-select-multiple-remove-btn {
```

```
font-size: 16px !important;  
}
```

```
.search-cd-cont {  
  display: flex;  
  justify-content: flex-end;  
  width: 80%;  
  gap: 0.5rem;  
  position: relative;  
  height: 35px;  
}
```

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": "^2.0.0"
```