

Website

edu-tracker/

- ├─ package.json
- ├─ tsconfig.json
- ├─ vite.config.ts
- ├─ index.html
- ├─ postcss.config.cjs
- ├─ tailwind.config.cjs
- ├─ README.md
- └─ src/
 - ├─ main.tsx
 - ├─ App.tsx
 - └─ index.css

1. package.json

json

Copy code

```
{
  "name": "edu-tracker",
  "version": "0.1.0",
  "private": true,
  "scripts": {
    "dev": "vite",
    "build": "vite build",
    "preview": "vite preview"
  },
  "dependencies": {
    "react": "^18.2.0",
    "react-dom": "^18.2.0",
    "lucide-react": "^0.259.0"
  },
  "devDependencies": {
    "typescript": "^5.5.0",
    "vite": "^5.1.0",
    "tailwindcss": "^4.4.0",
    "postcss": "^8.4.0",
    "autoprefixer": "^10.4.0",
    "@types/react": "^18.2.21",
```

```
"@types/react-dom": "^18.2.7"
}
}
```

2. tsconfig.json

json

Copy code

```
{
  "compilerOptions": {
    "target": "ES2020",
    "useDefineForClassFields": true,
    "lib": ["DOM", "ESNext"],
    "jsx": "react-jsx",
    "module": "ESNext",
    "moduleResolution": "Node",
    "strict": true,
    "esModuleInterop": true,
    "skipLibCheck": true,
    "forceConsistentCasingInFileNames": true,
    "resolveJsonModule": true,
    "isolatedModules": true,
    "noEmit": true
  },
  "include": ["src"]
}
```

3. vite.config.ts

ts

Copy code

```
import { defineConfig } from 'vite';
import react from '@vitejs/plugin-react';
```

```
export default defineConfig({
  plugins: [react()],
  server: {
    port: 5173
  }
});
```

If you don't have `@vitejs/plugin-react` installed, Vite still works for TSX but plugin is recommended. If you prefer simpler setup, you can omit plugin import and the plugin line — Vite will still handle TSX.

4. index.html

html

Copy code

5. postcss.config.cjs

js

Copy code

```
module.exports = {  
  plugins: [  
    tailwindcss,   
    autoprefixer,   
  ],  
};
```

6. tailwind.config.cjs

js

Copy code

```
module.exports = {  
  content: [  
    "./index.html",  
    "./src/**/*.ts",  
    "./src/**/*.tsx",  
    "./src/**/*.js",  
    "./src/**/*.jsx",  
  ],  
  theme: {  
    extend: {},  
  },  
  plugins: [],  
}
```

7. README.md

md

Copy code

EduTracker (Vite + React + TypeScript + Tailwind)

Setup

1. Install dependencies:

```
npm install
```

markdown

Copy code

2. Start dev server:

```
npm run dev
```

pgsql

Copy code

3. Open `http://localhost:5173` in the browser.

Notes

- This project uses Tailwind CSS classes in `src/index.css`.
- Icons are from `lucide-react`. If you want other icons, add them to the import in `src/App.tsx`.
- Data is static sample data. Connect to a backend or add persistence for real use.

8. `src/main.tsx`

tsx

Copy code

```
import React from 'react';
import { createRoot } from 'react-dom/client';
import './index.css';
import App from './App';
```

```
createRoot(document.getElementById('root') as HTMLElement).render(
  <React.StrictMode>
```

```
</React.StrictMode>
```

```
);
```

9) `src/index.css`

css

Copy code

```
@tailwind base;
@tailwind components;
@tailwind utilities;
```

/ small local helpers to match spacing & layout used in the App /

```
html, body, #root {
  height: 100%;
}
body {
  background-color: #F3F4F6; / gray-100 /
}
```

10) `src/App.tsx` (full)

This is the main app — paste it verbatim into `src/App.tsx`. It is TypeScript React and uses

lucide-react icons. I cleaned and unified the multiple fragments you provided and ensured TypeScript correctness.

tsx

Copy code

```
import React, { useState } from 'react';
```

```
import {
```

```
  LayoutDashboard,
```

```
  Users,
```

```
  BookOpen,
```

```
  BarChart3,
```

```
  Calendar,
```

```
  Bell,
```

```
  Settings,
```

```
  User,
```

```
  LogOut,
```

```
  Search,
```

```
  Filter,
```

```
  ChevronDown,
```

```
  Plus,
```

```
  Eye,
```

```
  Edit,
```

```
  Trash2,
```

```
  Check,
```

```
  X,
```

```
} from 'lucide-react';
```

```
type Performance = 'Excellent' | 'Good' | 'Average' | 'Needs Improvement';
```

```
type Status = 'Active' | 'Inactive';
```

```
interface Student {
```

```
  id: string;
```

```
  name: string;
```

```
  email: string;
```

```
  grade: string;
```

```
  attendance: number;
```

```
  performance: Performance;
```

```
  status: Status;
```

```
  credits?: number;
```

```
  activities?: number;
```

```
  clubs?: string[];
```

```
skills?: string[];  
}
```

```
interface Activity {  
  id: string;  
  title: string;  
  description: string;  
  date: string;  
  type: string;  
  credits?: number;  
  category?: string;  
  status?: string;  
  verifiedBy?: string;  
}
```

```
const sampleStudents: Student[] = [  
  { id: '1', name: 'Rahul Sharma', email: 'rahul.sharma@example.com', grade: 'B.Tech CSE 3rd  
Year', attendance: 92, performance: 'Excellent', status: 'Active', credits: 145, activities: 28,  
clubs: ['Coding Club'], skills: ['React', 'Python'] },  
  { id: '2', name: 'Priya Patel', email: 'priya.patel@example.com', grade: 'MBA 2nd Year',  
attendance: 88, performance: 'Good', status: 'Active', credits: 120, activities: 22, clubs: ['Debate  
Society'], skills: ['Marketing'] },  
  { id: '3', name: 'Aman Verma', email: 'aman.verma@example.com', grade: 'B.Sc 2nd Year',  
attendance: 74, performance: 'Average', status: 'Active', credits: 98, activities: 10, clubs: ['NSS'],  
skills: ['Public Speaking'] },  
  { id: '4', name: 'Simran Kaur', email: 'simran.kaur@example.com', grade: 'B.Tech ECE 4th Year',  
attendance: 96, performance: 'Excellent', status: 'Active', credits: 160, activities: 35, clubs:  
['Robotics'], skills: ['Embedded Systems'] },  
];
```

```
const sampleActivities: Activity[] = [  
  { id: 'a1', title: 'International AI Conference', description: '3-day international conference on AI.',  
date: '2024-01-15', type: 'Conference', credits: 5, category: 'Academic', status: 'Approved',  
verifiedBy: 'Dr. Amit Kumar' },  
  { id: 'a2', title: 'Google Cloud Certification', description: 'Completed Google Cloud Associate  
Engineer.', date: '2024-01-10', type: 'Certification', credits: 8, category: 'Technical', status:  
'Approved', verifiedBy: 'Prof. Sunita Singh' },  
  { id: 'a3', title: 'NSS Community Service', description: 'Blood donation camp and tree plantation.',  
date: '2023-12-05', type: 'Community Service', credits: 6, category: 'Social', status: 'Pending  
Approval', verifiedBy: '' },  
  { id: 'a4', title: 'Hackathon Winner', description: '1st prize in national hackathon.', date: '2023-11-  
20', type: 'Competition', credits: 12, category: 'Technical', status: 'Approved', verifiedBy: 'Dr. Anil
```

```
Gupta' },
{ id: 'a5', title: 'Coursera ML Specialization', description: 'Completed Machine Learning
specialization.', date: '2023-10-15', type: 'Online Course', credits: 7, category: 'Academic',
status: 'Approved', verifiedBy: 'Prof. Neha Verma' },
];
```

```
const App: React.FC = () => {
  const [activeTab, setActiveTab] = useState<'dashboard' | 'students' | 'activities' | 'reports'>
('dashboard');
  const [sidebarOpen, setSidebarOpen] = useState(true);
  const [showNotifications, setShowNotifications] = useState(false);
  const [showUserMenu, setShowUserMenu] = useState(false);
  const [userRole, setUserRole] = useState<'student' | 'faculty' | 'admin'>('admin');

  const [students] = useState<Student[]>(sampleStudents);
  const [activities] = useState<Activity[]>(sampleActivities);

  const pendingApprovals = activities.filter(a => a.status = 'Pending Approval' || a.status =
'Pending');

  const stats = {
    totalStudents: students.length,
    averageAttendance: Math.round(students.reduce((s, a) => s + a.attendance, 0) / Math.max(1,
students.length)),
    totalCredits: activities.reduce((s, a) => s + (a.credits || 0), 0),
    pendingApprovals: pendingApprovals.length,
    activityCategories: {
      academic: activities.filter(a => a.category = 'Academic').length,
      technical: activities.filter(a => a.category = 'Technical').length,
      extracurricular: activities.filter(a => a.category = 'Extracurricular').length,
      social: activities.filter(a => a.category = 'Social').length
    } as Record<string, number>
  };
};
```

```
/ -----
Render helpers
----- /
```

```
const renderDashboard = () => (
```

```
<StatCard label="Total Students" value={stats.totalStudents} icon={} />
```

```

<StatCard label="Average Attendance" value={ {{stats.averageAttendance}}% } icon={} />
<StatCard label="Total Credits Earned" value={stats.totalCredits} icon={} />
<StatCard label="Pending Approvals" value={stats.pendingApprovals} icon={} />

```

```

<div className="grid grid-cols-1 lg:grid-cols-2 gap-6">
  <div className="bg-white rounded-lg shadow p-6">
    <div className="flex items-center justify-between mb-6">
      <h3 className="text-lg font-semibold text-gray-900">Recent
Activities</h3>
      <button className="text-blue-600 text-sm font-medium">View
All</button>
    </div>
    <div className="space-y-4">
      {activities.slice(0, 5).map((activity) => (
        <div key={activity.id} className="flex items-center justify-between
p-3 bg-gray-50 rounded-lg">
          <div className="flex items-center space-x-3">
            <div className={`w-10 h-10 rounded-full flex items-center
justify-center ${
              activity.category === 'Academic' ? 'bg-blue-100' :
              activity.category === 'Technical' ? 'bg-purple-100' :
              activity.category === 'Extracurricular' ? 'bg-green-100' :
              'bg-yellow-100'
            }`}>
              <BookOpen className="w-5 h-5 text-blue-600" />
            </div>
            <div className="flex-1">
              <p className="font-medium text-gray-900">{activity.title}</p>
              <p className="text-sm text-gray-600">{activity.description}
</p>
              <div className="flex items-center mt-1 space-x-2">
                <span className="text-xs text-gray-500">{activity.date}
</span>
                <span className="text-xs font-medium text-blue-600">
{activity.credits} credits</span>
                <span className={`px-1 py-0.5 rounded text-xs ${
                  activity.status === 'Approved' ? 'bg-green-100 text-green-
800' :
                  'bg-yellow-100 text-yellow-800'
                }`}>
                  {activity.status}
                </span>
              </div>
            </div>
          </div>
        </div>
      )
    </div>
  </div>

```



```

    }}
  </div>
</div>

<div className="bg-white rounded-lg shadow p-6">
  <div className="flex items-center justify-between mb-6">
    <h3 className="text-lg font-semibold text-gray-900">Activity
Categories</h3>
    <button className="text-blue-600 text-sm font-medium">View
Analytics</button>
  </div>
  <div className="space-y-4">
    {Object.entries(stats.activityCategories).map(([category, count]) => (
      <div key={category} className="flex items-center justify-between">
        <span className="text-sm font-medium text-gray-700 capitalize">
{category}</span>
        <div className="flex items-center space-x-2">
          <div className="w-32 bg-gray-200 rounded-full h-2">
            <div
              className={`h-2 rounded-full ${
                category === 'academic' ? 'bg-blue-600' :
                category === 'technical' ? 'bg-purple-600' :
                category === 'extracurricular' ? 'bg-green-600' :
                'bg-yellow-600'
              }`}
              style={{ width: `${Math.min(100, (count / Math.max(1,
activities.length)) * 100)}%` }}
            </div>
          </div>
          <span className="text-sm text-gray-600 w-8 text-right">{count}
</span>
        </div>
      </div>
    ))}
  </div>
</div>

{ (userRole === 'faculty' || userRole === 'admin') && (
  <div className="bg-white rounded-lg shadow p-6">
    <div className="flex items-center justify-between mb-6">
      <h3 className="text-lg font-semibold text-gray-900">Approval
Queue</h3>
      <button className="text-blue-600 text-sm font-medium">Process
All</button>
    </div>
  </div>

```

```

    <div className="space-y-4">
      {pendingApprovals.slice(0, 3).map((activity) => (
        <div key={activity.id} className="flex items-center justify-between
p-3 bg-gray-50 rounded-lg">
          <div className="flex-1">
            <p className="font-medium text-gray-900">{activity.title}</p>
            <p className="text-sm text-gray-600">{activity.description}</p>
            <p className="text-xs text-gray-500 mt-1">Submitted by: Student
ID-{activity.id}</p>
          </div>
          <div className="flex items-center space-x-2">
            <button className="px-3 py-1 bg-green-100 text-green-800 text-sm
rounded-lg hover:bg-green-200">
              Approve
            </button>
            <button className="px-3 py-1 bg-red-100 text-red-800 text-sm
rounded-lg hover:bg-red-200">
              Reject
            </button>
          </div>
        </div>
      ))}
      {pendingApprovals.length > 3 && (
        <div className="text-center pt-2">
          <button className="text-blue-600 text-sm font-medium">
            View all {pendingApprovals.length} pending approvals
          </button>
        </div>
      )}
    </div>
  </div>
)}
</div>

```

);

const renderStudents = () => (

Student Profiles

Comprehensive student activity records and analytics

Add Record

```
<div className="bg-white rounded-lg shadow">
  <div className="p-4 border-b border-gray-200">
    <div className="flex flex-col sm:flex-row gap-4">
      <div className="relative flex-1">
        <Search className="absolute left-3 top-1/2 transform -translate-y-
1/2 text-gray-400 w-4 h-4" />
        <input
          type="text"
          placeholder="Search students by name, ID, or activity..."
          className="w-full pl-10 pr-4 py-2 border border-gray-300 rounded-
lg focus:ring-2 focus:ring-blue-500 focus:border-transparent"
        />
      </div>
      <select className="px-4 py-2 border border-gray-300 rounded-lg
focus:ring-2 focus:ring-blue-500 focus:border-transparent">
        <option>All Departments</option>
        <option>Computer Science</option>
        <option>Business Administration</option>
      </select>
    </div>
  </div>

  <div className="overflow-x-auto">
    <table className="w-full">
      <thead className="bg-gray-50">
        <tr>
          <th className="px-6 py-3 text-left text-xs font-medium text-gray-
500 uppercase tracking-wider">Student</th>
          <th className="px-6 py-3 text-left text-xs font-medium text-gray-
500 uppercase tracking-wider">Program & Year</th>
          <th className="px-6 py-3 text-left text-xs font-medium text-gray-
500 uppercase tracking-wider">Activities</th>
          <th className="px-6 py-3 text-left text-xs font-medium text-gray-
500 uppercase tracking-wider">Credits</th>
          <th className="px-6 py-3 text-left text-xs font-medium text-gray-
500 uppercase tracking-wider">Performance</th>
```

```

        <th className="px-6 py-3 text-left text-xs font-medium text-gray-
500 uppercase tracking-wider">Status</th>
        <th className="px-6 py-3 text-left text-xs font-medium text-gray-
500 uppercase tracking-wider">Actions</th>
    </tr>
</thead>
<tbody className="bg-white divide-y divide-gray-200">
    {students.map((student) => (
        <tr key={student.id}>
            <td className="px-6 py-4 whitespace-nowrap">
                <div className="flex items-center">
                    <div className="w-10 h-10 bg-blue-100 rounded-full flex
items-center justify-center mr-3">
                        <User className="w-5 h-5 text-blue-600" />
                    </div>
                    <div>
                        <div className="text-sm font-medium text-gray-900">
{student.name}</div>
                        <div className="text-sm text-gray-500">{student.email}
</div>
                    </div>
                </td>
                <td className="px-6 py-4 whitespace-nowrap text-sm text-gray-
900">{student.grade}</td>
                <td className="px-6 py-4 whitespace-nowrap">
                    <span className="text-sm text-gray-900">{student.activities}
</span>
                </td>
                <td className="px-6 py-4 whitespace-nowrap">
                    <span className="px-2 py-1 bg-blue-100 text-blue-800 text-xs
font-medium rounded-full">
                        {student.credits} credits
                    </span>
                </td>
                <td className="px-6 py-4 whitespace-nowrap">
                    <span className={`px-2 py-1 rounded-full text-xs font-medium
${
                    student.performance === 'Excellent' ? 'bg-green-100 text-
green-800' :
                    student.performance === 'Good' ? 'bg-blue-100 text-blue-800'
:
                    student.performance === 'Average' ? 'bg-yellow-100 text-
yellow-800' :
                    'bg-red-100 text-red-800'
                }`}>

```

```

        {student.performance}
      </span>
    </td>
    <td className="px-6 py-4 whitespace-nowrap">
      <span className={`px-2 py-1 rounded-full text-xs font-medium
${
      student.status === 'Active' ? 'bg-green-100 text-green-800'
: 'bg-gray-100 text-gray-800'
} `}>
        {student.status}
      </span>
    </td>
    <td className="px-6 py-4 whitespace-nowrap text-sm text-gray-
500">
      <div className="flex items-center space-x-2">
        <button className="text-blue-600 hover:text-blue-800"
title="View Portfolio">
          <Eye className="w-4 h-4" />
        </button>
        <button className="text-green-600 hover:text-green-800"
title="Edit Profile">
          <Edit className="w-4 h-4" />
        </button>
        <button className="text-red-600 hover:text-red-800"
title="Delete">
          <Trash2 className="w-4 h-4" />
        </button>
      </div>
    </td>
  </tr>
</tbody>
</table>
</div>
</div>
</div>

```

);

const renderActivities = () => (

Activity Tracker

Manage and track all student activities and achievements

Log New Activity

```
<div className="grid grid-cols-1 md:grid-cols-4 gap-4 mb-6">
  <SmallCard label="Total Activities" value={activities.length} />
  <SmallCard label="Approved" value={activities.filter(a => a.status ===
'Approved').length} />
  <SmallCard label="Pending" value={activities.filter(a => a.status ===
'Pending Approval' || a.status === 'Pending').length} />
  <SmallCard label="Total Credits" value={activities.reduce((sum, a) => sum
+ (a.credits || 0), 0)} />
</div>

<div className="bg-white rounded-lg shadow">
  <div className="p-4 border-b border-gray-200">
    <div className="flex flex-col sm:flex-row gap-4">
      <div className="relative flex-1">
        <Search className="absolute left-3 top-1/2 transform -translate-y-
1/2 text-gray-400 w-4 h-4" />
        <input
          type="text"
          placeholder="Search activities..."
          className="w-full pl-10 pr-4 py-2 border border-gray-300 rounded-
lg focus:ring-2 focus:ring-blue-500 focus:border-transparent"
        />
      </div>
      <select className="px-4 py-2 border border-gray-300 rounded-lg">
        <option>All Types</option>
        <option>Conference</option>
        <option>Certification</option>
        <option>Competition</option>
      </select>
      <select className="px-4 py-2 border border-gray-300 rounded-lg">
        <option>All Status</option>
        <option>Approved</option>
        <option>Pending Approval</option>
      </select>
    </div>
  </div>
</div>

<div className="overflow-x-auto">
```

```

<table className="w-full">
  <thead className="bg-gray-50">
    <tr>
      <th className="px-6 py-3 text-left text-xs font-medium text-gray-500 uppercase tracking-wider">Activity</th>
      <th className="px-6 py-3 text-left text-xs font-medium text-gray-500 uppercase tracking-wider">Type</th>
      <th className="px-6 py-3 text-left text-xs font-medium text-gray-500 uppercase tracking-wider">Date</th>
      <th className="px-6 py-3 text-left text-xs font-medium text-gray-500 uppercase tracking-wider">Credits</th>
      <th className="px-6 py-3 text-left text-xs font-medium text-gray-500 uppercase tracking-wider">Status</th>
      <th className="px-6 py-3 text-left text-xs font-medium text-gray-500 uppercase tracking-wider">Verified By</th>
      <th className="px-6 py-3 text-left text-xs font-medium text-gray-500 uppercase tracking-wider">Actions</th>
    </tr>
  </thead>
  <tbody className="bg-white divide-y divide-gray-200">
    {activities.map((activity) => (
      <tr key={activity.id}>
        <td className="px-6 py-4">
          <div>
            <div className="text-sm font-medium text-gray-900">
{activity.title}</div>
            <div className="text-sm text-gray-500">
{activity.description}</div>
          </div>
        </td>
        <td className="px-6 py-4 whitespace-nowrap">
          <span className={`px-2 py-1 rounded-full text-xs font-medium
${
      activity.type === 'Conference' ? 'bg-blue-100 text-blue-800'
:
      activity.type === 'Certification' ? 'bg-purple-100 text-
purple-800' :
      activity.type === 'Community Service' ? 'bg-yellow-100 text-
yellow-800' :
      'bg-gray-100 text-gray-800'
}`}>
            {activity.type}
          </span>
        </td>
        <td className="px-6 py-4 whitespace-nowrap text-sm text-gray-900">{activity.date}</td>
      </tr>
    ))}
  </tbody>
</table>

```

```

        <td className="px-6 py-4 whitespace-nowrap">
            <span className="px-2 py-1 bg-blue-100 text-blue-800 text-xs
font-medium rounded-full">
                {activity.credits}
            </span>
        </td>
        <td className="px-6 py-4 whitespace-nowrap">
            <span className={`px-2 py-1 rounded-full text-xs font-medium
${
                activity.status === 'Approved' ? 'bg-green-100 text-green-
800' : 'bg-yellow-100 text-yellow-800'
            }`}>
                {activity.status}
            </span>
        </td>
        <td className="px-6 py-4 whitespace-nowrap text-sm text-gray-
500">
            {activity.verifiedBy || '-'}
        </td>
        <td className="px-6 py-4 whitespace-nowrap text-sm text-gray-
500">
            <div className="flex items-center space-x-2">
                <button className="text-blue-600 hover:text-blue-800">
                    <Eye className="w-4 h-4" />
                </button>
                <button className="text-green-600 hover:text-green-800">
                    <Edit className="w-4 h-4" />
                </button>
                { (userRole === 'faculty' || userRole === 'admin') &&
activity.status !== 'Approved' ? (
                    <>
                        <button className="text-green-600 hover:text-green-800"
title="Approve"><Check className="w-4 h-4" /></button>
                        <button className="text-red-600 hover:text-red-800"
title="Reject"><X className="w-4 h-4" /></button>
                    </>
                ) : null }
            </div>
        </td>
    </tr>
</tbody>
</table>
</div>
</div>
</div>

```



```
);
```

```
const renderReports = () => (
```

Analytics & Reports

Generate comprehensive reports for accreditation and evaluation

Generate Report

```
<div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-6">
  <ReportCard title="NAAC Reports" description="Generate reports for NAAC
accreditation requirements" />
  <ReportCard title="AICTE Reports" description="Create reports for AICTE
compliance and submissions" />
  <ReportCard title="NIRF Reports" description="Prepare data for NIRF
rankings and submissions" />
  <ReportCard title="Student Portfolio" description="Export digital
portfolio for students" variant="green" />
  <ReportCard title="Department Analytics" description="Department-wise
performance and activity reports" variant="purple" />
  <ReportCard title="Custom Reports" description="Create customized reports
with specific parameters" variant="gray" />
</div>
</div>
```

```
);
```

```
const renderContent = () => {
switch (activeTab) {
case 'dashboard': return renderDashboard();
case 'students': return renderStudents();
case 'activities': return renderActivities();
case 'reports': return renderReports();
```

```
default: return renderDashboard();  
}  
};
```

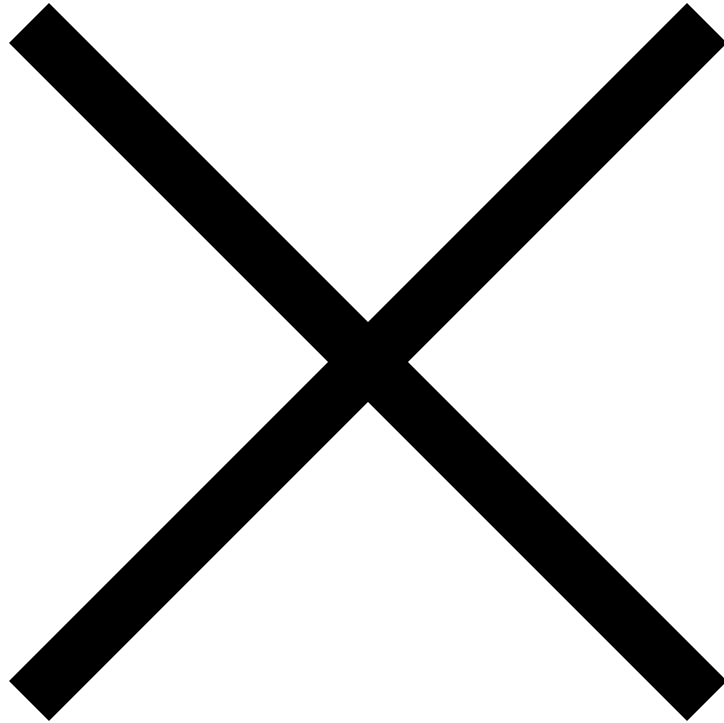
```
return (
```

```
{/ Sidebar /}
```

```
<div className={fixed inset-y-0 left-0 z-50 w-64 bg-white shadow-lg transform  
transition-transform duration-200 ${sidebarOpen ? 'translate-x-0' : '-translate-x-  
full'} lg:translate-x-0 }>
```

EduTracker

```
<button onClick={() => setSidebarOpen(false)} className="lg:hidden text-gray-400 hover:text-  
gray-600">
```



```
<nav className="p-4">
  <div className="space-y-1">
    {[
      { id: 'dashboard', icon: LayoutDashboard, label: 'Dashboard' },
      { id: 'students', icon: Users, label: 'Students' },
      { id: 'activities', icon: BookOpen, label: 'Activities' },
      { id: 'reports', icon: BarChart3, label: 'Reports' },
    ]}.map((item) => (
      <button key={item.id} onClick={() => setActiveTab(item.id as any)}
        className={`w-full flex items-center space-x-3 px-4 py-3 rounded-lg
          transition-colors ${activeTab === item.id ? 'bg-blue-100 text-blue-700' :
            'text-gray-700 hover:bg-gray-100'}}`>
```

```

        <item.icon className="w-5 h-5" />
        <span className="font-medium">{item.label}</span>
      </button>
    )}]
  </div>

  <div className="mt-8 pt-8 border-t border-gray-200">
    <div className="mb-3 px-1">
      <label className="text-xs text-gray-500">Role</label>
      <select value={userRole} onChange={(e) => setUserRole(e.target.value
as any)} className="mt-2 block w-full text-sm border border-gray-200 rounded
px-2 py-2">
        <option value="student">Student</option>
        <option value="faculty">Faculty</option>
        <option value="admin">Admin</option>
      </select>
    </div>

    {[ { id: 'settings', icon: Settings, label: 'Settings' }, { id:
'logout', icon: LogOut, label: 'Logout' } ].map((item) => (
      <button key={item.id} className="w-full flex items-center space-x-3
px-4 py-3 rounded-lg text-gray-700 hover:bg-gray-100 transition-colors">
        <item.icon className="w-5 h-5" />
        <span className="font-medium">{item.label}</span>
      </button>
    )]}
  </div>
</nav>
</div>

{/* Main content */}
<div className={`lg:pl-64 transition-spacing duration-200 ${sidebarOpen ?
'lg:pl-64' : 'lg:pl-0'}`}>
  {/* Header */}
  <header className="bg-white shadow-sm border-b border-gray-200">
    <div className="flex items-center justify-between px-6 py-4">
      <div className="flex items-center space-x-4">
        <button onClick={() => setSidebarOpen(true)} className="lg:hidden
text-gray-400 hover:text-gray-600">
          <svg className="w-6 h-6" fill="none" stroke="currentColor"
viewBox="0 0 24 24"><path strokeLinecap="round" strokeLinejoin="round"
strokeWidth={2} d="M4 6h16M4 12h16M4 18h16"/></svg>
        </button>
        <h1 className="text-xl font-semibold text-gray-900 capitalize">
{activeTab}</h1>
      </div>

```

```

    <div className="flex items-center space-x-4">
      <div className="relative">
        <button onClick={() => setShowNotifications(!showNotifications)}
className="p-2 text-gray-400 hover:text-gray-600 relative">
          <Bell className="w-5 h-5" />
          <span className="absolute top-1 right-1 w-2 h-2 bg-red-500
rounded-full"></span>
        </button>
      </div>
      <div className="relative">
        <button onClick={() => setShowUserMenu(!showUserMenu)}
className="flex items-center space-x-2 p-2 hover:bg-gray-100 rounded-lg">
          <div className="w-8 h-8 bg-blue-600 rounded-full flex items-
center justify-center">
            <User className="w-4 h-4 text-white" />
          </div>
          <span className="text-sm font-medium text-gray-900">Admin</span>
          <ChevronDown className="w-4 h-4 text-gray-400" />
        </button>
      </div>
    </div>
  </div>
</header>

  {/* Main content area */}
  <main className="p-6">
    {renderContent()}
  </main>
</div>

  {/* Backdrop for mobile sidebar */}
  {sidebarOpen && (
    <div className="fixed inset-0 bg-black bg-opacity-50 z-40 lg:hidden"
onClick={() => setSidebarOpen(false)} />
  )}
</div>

```

```
);
```

```
};
```

```
/ ----- small UI components ----- /
```

```
const StatCard: React.FC<{ label: string; value: string | number; icon?: React.ReactNode }> = ({
label, value, icon }) => (
```

```
{label}
```

```
{value}
```

```
{icon}
```

```
);
```

```
const SmallCard: React.FC<{ label: string; value: number | string }> = ({ label, value }) => (
```

```
{label}
```

```
{value}
```

```
);
```

```
const ReportCard: React.FC<{ title: string; description: string; variant?: 'blue' | 'green' | 'purple' |  
'gray' }> = ({ title, description, variant = 'blue' }) => {
```

```
const colors = {
```

```
blue: 'bg-blue-600 hover:bg-blue-700',
```

```
green: 'bg-green-600 hover:bg-green-700',
```

```
purple: 'bg-purple-600 hover:bg-purple-700',
```

```
gray: 'bg-gray-600 hover:bg-gray-700'
```

```
} as const;
```

```
return (
```

{title}

```
{description}
```

```
<button className={w-full ${colors[variant]} text-white px-4 py-2 rounded-  
lg}>Generate
```

```
);
```

```
};
```

```
export default App;
```

Run instructions (repeat)

Create folder e.g. edu-tracker and place the files exactly as above.

Run:

bash

Copy code

npm install

npm run dev

Open the printed URL (default <http://localhost:5173/>).

Notes & extras I added

Tailwind CSS integration (index.css + config).

Role selector to change UI behavior (student/faculty/admin).

Approval queue & pending approvals rendering.

All icons used are imported from lucide-react — if you see unused icons warnings you can remove them from the import line.

Data is static; everything is in-memory to keep this example easy to run.

I kept layout and styling consistent with the fragments you provided and matching the look of the provided site.