TCET
DEPARTMENT OF COMPUTER ENGINEERING (COMP)
Choice Based Credit Grading Scheme with Holistic and Multidisciplinary Education
Under Autonomy - CBCGS-HME 2023
University of Mumbai

.

🎓 STUDENT-TEACHER PORTAL

MINI PROJECT
Industrial Practices
Technologies Used: ReactJS, CSS
GitHub Repository: *https://github.com/Vishal93727/Student-Teacher-Portal*

TEAM MEMBERS (GROUP B-23)

- **ADITYA MISHRA - 25**
- **SHLOK MISHRA - 26**
- **VISHAL MOURYA - 27**

🔹 Abstract

The Student-Teacher Portal is a responsive front-end application developed to create a seamless interaction between students and teachers. This system allows teachers to assign work and tests, while students can submit assignments and check grades. It features distinct dashboards for both roles, ensuring a personalized and organized academic experience. The current version is limited to the front-end and sets the foundation for a complete full-stack application.

🔹 Introduction

With the growth of online education and digital classrooms, there is a growing need for an efficient platform to manage academic interactions. Traditional methods of assignment distribution and grading are time-consuming. This project aims to provide a modern digital solution that simplifies the workflow for both teachers and students.

🔹 Problem Statement

Manual handling of academic tasks leads to delays, miscommunication, and inefficiencies. Students often miss important deadlines, and teachers find it difficult to organize submissions and grades. A centralized digital system can improve communication and productivity in educational environments.

🔹 Features / Modules

Student Features:

- View and submit assignments
- Track deadlines and upcoming tests
- View recent grades

Teacher Features:

- Create and assign assignments
- Create tests using forms
- Grade student submissions
- View student performance

Common Features:

- Role-based login (Student/Teacher)
- Responsive UI for all devices
- Clean dashboard interface
- Ready for file upload integration via Multer (in backend phase)
- Pagination and filtering support for large data handling (planned for backend)

🔹 Technologies Used

- **Frontend:** ReactJS (for component-based design), CSS (for layout and responsiveness)
- **Backend (Planned):** Node.js, Express.js, MongoDB
- **File Uploads (Planned):** Multer
- **Version Control:** Git and GitHub

🔹 Implementation

The project is divided into two primary user roles:

1. Student Interface:
Displays pending assignments, test schedules, and grades. Students can view details of tasks and interact with a clean UI that is mobile-friendly and responsive.

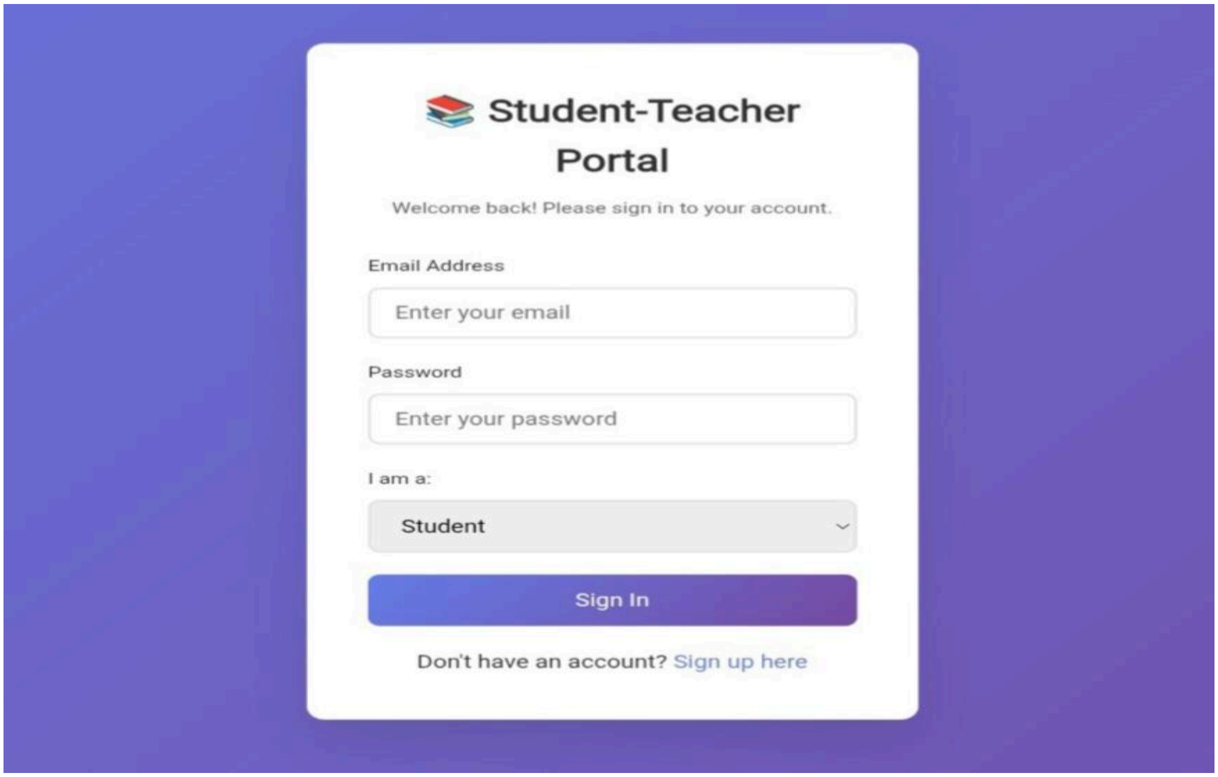
2. Teacher Interface:
Teachers have a dashboard that displays total assignments, pending reviews, and student submissions. There is an option to create new tests using forms and review submissions.

Development Notes:

- The UI is developed using React functional components and hooks.
- Styling and layout were handled entirely through custom CSS.
- Navigation was structured through role-based routes for clean user separation.

RESULT AND DISCUSSION

1. **Login/Register Page** (with role selection)



📚 Student-Teacher Portal

Welcome back! Please sign in to your account.

Email Address

Enter your email

Password

Enter your password

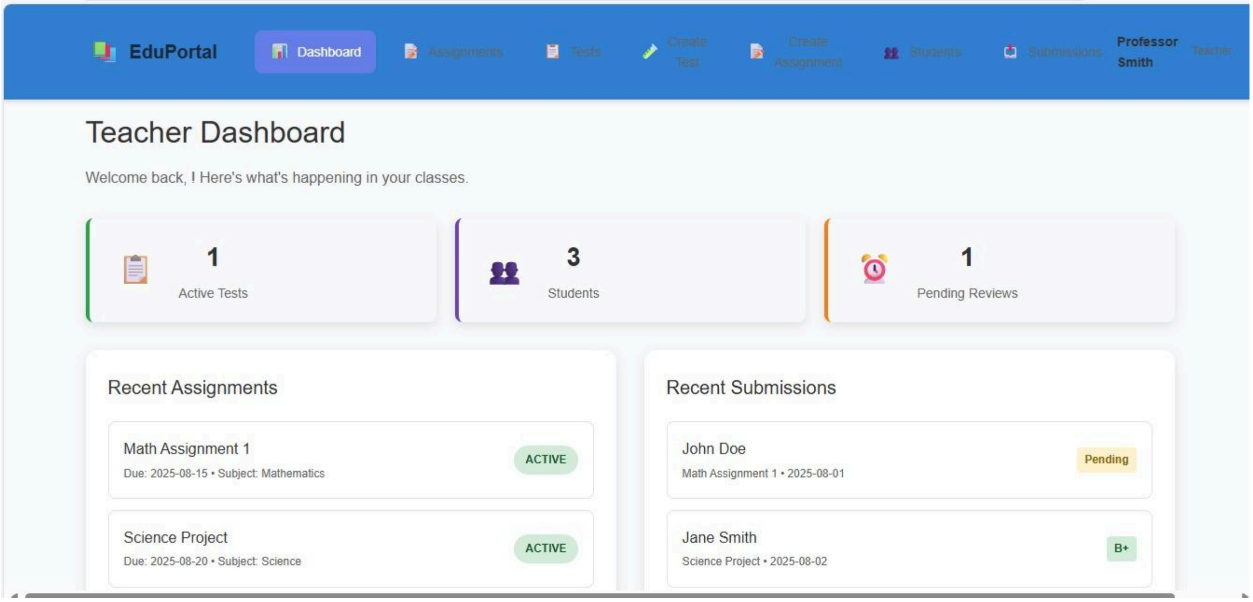
I am a:

Student

Sign In

Don't have an account? Sign up here

2. **Teacher Dashboard**



Teacher Dashboard

Welcome back, ! Here's what's happening in your classes.

1 Active Tests

3 Students

1 Pending Reviews

Recent Assignments

Math Assignment 1 — ACTIVE
Due: 2025-08-15 • Subject: Mathematics

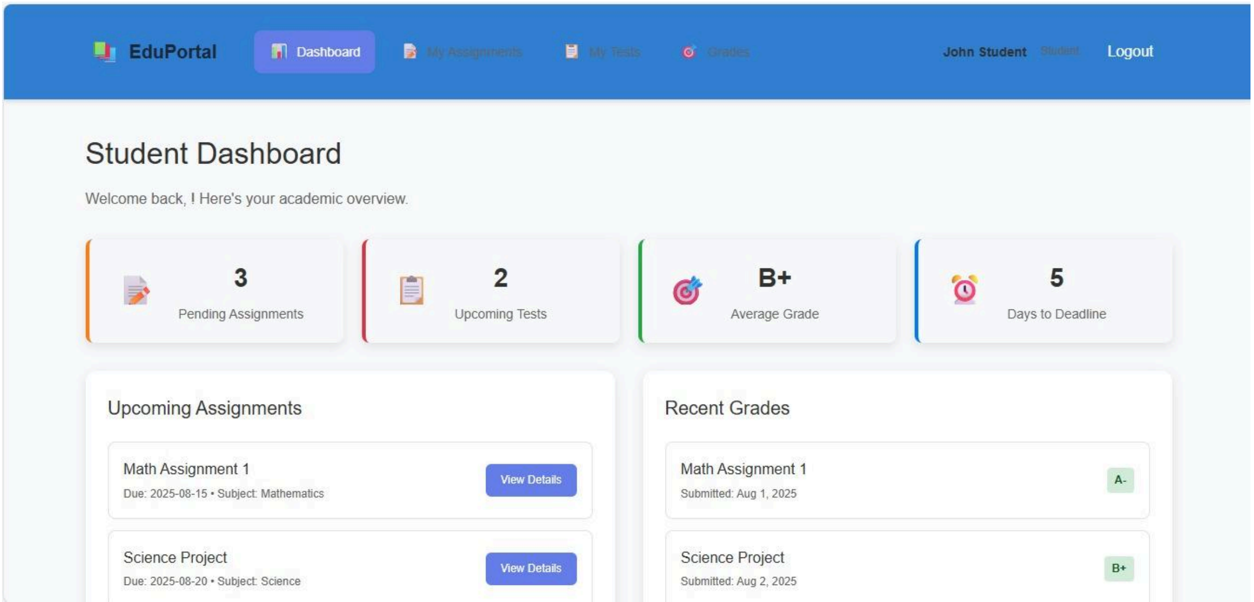
Science Project — ACTIVE
Due: 2025-08-20 • Subject: Science

Recent Submissions

John Doe — Pending
Math Assignment 1 • 2025-08-01

Jane Smith — B+
Science Project • 2025-08-02

3. **Student Dashboard**



Student Dashboard

Welcome back, ! Here's your academic overview.

3	Pending Assignments
2	Upcoming Tests
B+	Average Grade
5	Days to Deadline

Upcoming Assignments

Math Assignment 1
Due: 2025-08-15 • Subject: Mathematics
View Details

Science Project
Due: 2025-08-20 • Subject: Science
View Details

Recent Grades

Math Assignment 1
Submitted: Aug 1, 2025
A-

Science Project
Submitted: Aug 2, 2025
B+

◆ Conclusion

This front-end prototype achieves its objective of digitizing basic student-teacher interactions. The clean interface and organized modules make it scalable and efficient. It sets the groundwork for a complete full-stack application to be built in the next phase.

◆ Future Scope

In the upcoming development phase, the following functionalities are planned:

- **User Authentication:** Using JWT for secure logins
- **Backend Integration:** Node.js, Express.js with MongoDB to store users, assignments, and grades
- **File Uploads:** Students will be able to submit PDF/DOC files via Multer
- **Test Management:** Create and attempt online tests, including auto-grading
- **Email Notifications:** Send deadline reminders or grade updates
- **Search and Filter System:** Enhance usability for teachers with many students
- **Analytics Dashboard:** For tracking student performance
- **Feedback System:** Allow comments and suggestions on submissions
- **PWA Support:** Make the app installable and usable offline
- **Hosting:** Vercel (frontend), Render/Railway (backend).

🔹 References

- ReactJS Documentation
- CSS Tricks
- Multer File Upload
- MongoDB Docs

🔹 GitHub Repository

GitHub repo link here:
👉 https://github.com/Vishal93727/Student-Teacher-Portal

🔹 ANNEXURE

```
import React, { useState, useEffect } from
'react';
import LoginComponent from
'../components/LoginComponent';
import Navigation from
'../components/Navigation';
import StudentDashboard from
'./StudentDashboard';
import TeacherDashboard from
'./TeacherDashboard';
import TestBuilder from './TestBuilder';
import ComingSoon from
'./ComingSoon';import ProfilePage from
'./ProfilePage';
import StudentAssignments from
'./StudentAssignments';
import TestsPage from './TestsPage';
import GradesPage from './GradesPage';
import SubmissionsPage from
'./SubmissionsPage';

import Assignment from
'../pages/Assignment';
import Register from
'../components/Register';
import Navbar from
```

```
sub.studentName ===
currentUser.name)}

          />
        );
      case 'test-builder':
        return currentUser.role === 'teacher'
? <TestBuilder /> : <div>Access
Denied</div>;
      case 'create-assignment':
        return currentUser.role === 'teacher'
? <Assignment /> : <div>Access
Denied</div>;
      case 'register':
        return <Register
onRegistrationSuccess={handleLogin} />;
      case 'profile':
        return <div><ProfilePage/></div>;
      case 'assignments':
        return currentUser.role === 'teacher'
? (
          <Assignment
assignments={assignments} />
        ) : (
          <div><StudentAssignments/></div>
        );
```

```
'../components/Navbar'
import TeacherStudentFilter from
'./TeacherStudentFilter';
import { mockAssignments, mockTests,
mockStudents, mockSubmissions } from
'../data/mockData';
import '../styles/styles.css';

const StudentTeacherPortal = () => {
  const [currentUser, setCurrentUser] =
useState(null);
  const [currentView, setCurrentView] =
useState('login');
  const [assignments, setAssignments] =
useState([]);
    const [createAssignments,
setCreateAssignments] = useState([]);

  const [tests, setTests] = useState([]);
  const [students, setStudents] =
useState([]);
  const [submissions, setSubmissions] =
useState([]);
  const [isMobileMenuOpen,
setIsMobileMenuOpen] = useState(false);

useEffect(() => {
  const fetchData = async () => {
    try {
      const [assignRes, testRes, subRes] =
await Promise.all([

fetch("http://localhost:5000/api/assignmen
ts").then(r => r.json()),

fetch("http://localhost:5000/api/tests").the
n(r => r.json()),

fetch("http://localhost:5000/api/submissio
ns").then(r => r.json())
      ]);
      setAssignments(assignRes);
      setTests(testRes);
      setSubmissions(subRes);
    } catch (err) {
```

```
      switch (currentView) {
      case 'dashboard':
        return currentUser.role === 'teacher'
? (
          <TeacherDashboard
            assignments={assignments}
            tests={tests}
            students={students}
            submissions={submissions}
 createAssignments={createAssignments}
          />
        ) : (
          <StudentDashboard
            assignments={assignments}
            tests={tests}
           submissions={submissions.filter(sub
=>
case 'tests':
        return currentUser.role === 'teacher'
? (
          <div>Tests Page - Coming
Soon</div>  ) : (
          <TestsPage/>     );
      case 'grades':
        return currentUser.role === 'teacher'
? (
          <div>Grades Page - Coming
Soon</div>
        ) : (
          <GradesPage/>     );
      case 'submissions':
        return currentUser.role === 'teacher'
? (
          <div>Submissions Page - Coming
Soon</div>
        ) : (
          <SubmissionsPage/>     );
      case 'logout':
        handleLogout();
        return <div>Logging out...</div>;
      case 'coming-soon':
        return <ComingSoon />;
      case 'login':
        return <LoginComponent
onLogin={handleLogin} />;
```

```
      console.error("Error loading data:",
err);
    }
  };
  fetchData();
}, []);

  const handleLogin = (user) => {
    setCurrentUser(user);
    setCurrentView('dashboard');
  };

  const handleLogout = () => {
    setCurrentUser(null);
    setCurrentView('login');
    setIsMobileMenuOpen(false);
  };

  const renderCurrentView = () => {
    if (!currentUser) return (<>
      <Navbar/>
    <LoginComponent
onLogin={handleLogin} />
    </>);
```

```
      case 'students':
        return <TeacherStudentFilter />;
        default:
        return <ComingSoon />;
    }  };
return (
    <div className="app">
      {currentUser && (
        <Navigation
          currentUser={currentUser}
          currentView={currentView}
          setCurrentView={setCurrentView}
          handleLogout={handleLogout}
isMobileMenuOpen={isMobileMenuOpent
oggleMobileMenu={setIsMobileMenuOpe
n}
        />  )}
      <main className="main-content">
        {renderCurrentView()}
      </main>
    </div>  );
};
export default StudentTeacherPortal;
```

```
import React, { useState, useEffect } from
'react';
const StudentDashboard = () => {
    const stats = [
      { title: 'Pending Assignments', value:
3, icon: '📝', color: 'orange' },
      { title: 'Upcoming Tests', value: 2, icon:
'📋', color: 'red' },
      { title: 'Average Grade', value: 'B+',
icon: '🎯', color: 'green' },
      { title: 'Days to Deadline', value: 5,
icon: '⏰', color: 'blue' }
    ];
  const [currentUser, setCurrentUser] =
useState(null);
  const [currentView, setCurrentView] =
useState('login');
```

```
        <div className="stats-grid">
          {stats.map((stat, index) => (
            <div key={index}
className={`stat-card ${stat.color}`}>
              <div
className="stat-icon">{stat.icon}</div>
              <div className="stat-content">
                <h3>{stat.value}</h3>
                <p>{stat.title}</p>
              </div>     </div>       ))}
        </div>
        <div
className="dashboard-content">
          <div className="recent-section">
            <h2>Upcoming
Assignments</h2>
            <div
```

```
  const [assignments, setAssignments] =
useState([]);
  const [tests, setTests] = useState([]);
  const [submissions, setSubmissions] =
useState([]);
  const [students, setStudents] =
useState([]);
  const [isMobileMenuOpen,
setIsMobileMenuOpen] = useState(false);
useEffect(() => {
  const studentId =
localStorage.getItem("studentId") ||
"<some-id>";
  const load = async () => {
    try {
      const res = await
fetch(`http://localhost:5000/api/dashboard
/student/${studentId}`);
      if (!res.ok) throw new Error("Failed to
load dashboard");
      const json = await res.json();
      setCurrentUser(json.student);
setAssignments(json.pendingAssignment
s); // you may want to keep all
assignments too
      setTests(json.upcomingTests);

setSubmissions(json.recentSubmissions);
 {pendingAssignmentsCount,
upcomingTestsCount, averageGrade,
daysToNearestDeadline}
    } catch (err) {
      console.error(err);
    } };
  load();
}, []);
    return (
      <div className="dashboard">
        <div
className="dashboard-header">
          <h1>Student Dashboard</h1>
          <p>Welcome back,
{currentUser?.name}! Here's your
academic overview.</p>
        </div>
```

```
className="assignment-list">
              {assignments.map(assignment
=> (
                <div key={assignment.id}
className="assignment-item">
                  <div
className="assignment-info">
                    <h4>{assignment.title}</h4>
                    <p>Due:
{assignment.dueDate} • Subject:
{assignment.subject}</p>
                  </div>
                  <button
className="submit-btn">View
Details</button>
                </div>           ))}     </div>
</div>
          <div className="recent-section">
            <h2>Recent Grades</h2>
            <div className="grade-list">
              <div className="grade-item">
                <div className="grade-info">
                  <h4>Math Assignment 1</h4>
                  <p>Submitted: Aug 1,
2025</p>
                </div>
                <span className="grade
graded">A-</span>
              </div>
              <div className="grade-item">
                <div className="grade-info">
                  <h4>Science Project</h4>
                  <p>Submitted: Aug 2,
2025</p>
                </div>
                <span className="grade
graded">B+</span>
              </div>
            </div>
          </div>
        </div>
      </div>
    );
  };
  export default StudentDashboard;
```

```
import Assignment from
'../pages/Assignment';

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


const TeacherDashboard = ({
assignments, tests, submissions,
students, currentUser }) => {
  const stats = [
    { title: 'Active Tests', value: tests.filter(t
=> t.status === 'Published').length, icon:
'📋', color: 'green' },
    { title: 'Students', value:
students.length, icon: '👥', color: 'purple'
},
    { title: 'Pending Reviews', value:
submissions.filter(s => !s.grade).length,
icon: '⏰', color: 'orange' }
  ];
const [data, setData] = useState({
assignments: [], submissions: [], students:
[], tests: [], stats: {} });

  useEffect(() => {
    const fetchDashboard = async () => {
      const res = await
fetch(`http://localhost:5000/api/dashboard
/teacher/${currentUser._id}`);
      const json = await res.json();
      setData(json);
    };
    fetchDashboard();
  }, [currentUser]);
  return (
    <div className="dashboard">
      <div
className="dashboard-header">
        <h1>Teacher Dashboard</h1>
        <p>Welcome back,
{currentUser?.name}! Here's what's
happening in your classes.</p>
```

```
<div className="dashboard-content">
        {/* Recent Assignments */}
        <div className="recent-section">
          <h2>Recent Assignments</h2>
          <div className="assignment-list">
            {assignments.slice(0,
3).map(assignment => (
              <div key={assignment.id}
className="assignment-item">
                <div
className="assignment-info">
                  <h4>{assignment.title}</h4>
                  <p>Due:
{assignment.dueDate} • Subject:
{assignment.subject}</p>
                </div>
                <span className={`status
${assignment.status.toLowerCase()}`}>{a
ssignment.status}</span>
              </div>
            ))}
          </div>
        </div>

        {/* Recent Submissions */}
        <div className="recent-section">
          <h2>Recent Submissions</h2>
          <div className="submission-list">
            {submissions.slice(0,
3).map(submission => (
              <div key={submission.id}
className="submission-item">
                <div
className="submission-info">

<h4>{submission.studentName}</h4>
                  <p>{submission.assignment} •
{submission.submittedAt}</p>
                </div>
                <span className={`grade
${submission.grade ? 'graded' :
'pending'}`}>
                  {submission.grade ||
      </div>

      <div className="stats-grid">
        {stats.map((stat, index) => (
          <div key={index} className={`stat-card ${stat.color}`}>
            <div className="stat-icon">{stat.icon}</div>
            <div className="stat-content">
              <h3>{stat.value}</h3>
              <p>{stat.title}</p>
            </div>
          </div>
        ))}
      </div>
```

```
'Pending'}
                </span>
              </div>
            ))}
          </div>
        </div>
      </div>
    </div>
  );
};

export default TeacherDashboard;
```