Student Activities - Frontend Code (React.js)

Project Structure

```
frontend/
├── public/
│   └── index.html
├── src/
│   ├── components/
│   │   ├── Layout/
│   │   │   ├── Navbar.jsx
│   │   │   └── Sidebar.jsx
│   │   ├── Notice/
│   │   │   ├── NoticeCard.jsx
│   │   │   ├── NoticeForm.jsx
│   │   │   └── NoticeList.jsx
│   │   ├── Admin/
│   │   │   ├── UserManagement.jsx
│   │   │   └── PrivilegeManager.jsx
│   │   └── ProtectedRoute.jsx
│   ├── context/
│   │   └── AuthContext.jsx
│   ├── pages/
│   │   ├── Login.jsx
│   │   ├── Register.jsx
│   │   ├── Dashboard.jsx
│   │   ├── AcademicNotices.jsx
│   │   ├── ClubActivities.jsx
│   │   ├── CreateNotice.jsx
│   │   └── AdminPanel.jsx
│   ├── services/
│   │   └── api.js
│   ├── App.jsx
│   ├── index.js
│   └── index.css
├── .env
└── package.json
```

1. package.json

```
{
  "name": "student-activities-frontend",
  "version": "1.0.0",
  "description": "Frontend for Student Activities Notice Board",
  "private": true,
  "dependencies": {
    "react": "^18.2.0",
    "react-dom": "^18.2.0",
    "react-router-dom": "^6.20.0",
    "axios": "^1.6.2",
    "react-icons": "^4.12.0"
  },
  "scripts": {
    "start": "react-scripts start",
    "build": "react-scripts build",
    "test": "react-scripts test",
    "eject": "react-scripts eject"
  },
  "devDependencies": {
    "react-scripts": "5.0.1"
  },
  "browserslist": {
    "production": [
      "&gt;0.2%",
      "not dead",
```

```
      "not op_mini all"
    ],
    "development": [
      "last 1 chrome version",
      "last 1 firefox version",
      "last 1 safari version"
    ]
  }
}
```

2. .env

```
REACT_APP_API_URL=http://localhost:5000/api
```

3. src/index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  &lt;React.StrictMode&gt;
    &lt;App /&gt;
  &lt;/React.StrictMode&gt;
);
```

4. src/index.css

```
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

body {
  font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen',
    'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',
    sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  background-color: #f5f5f5;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 20px;
}

.card {
  background: white;
  border-radius: 8px;
  padding: 20px;
  margin-bottom: 20px;
  box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
}

.btn {
  padding: 10px 20px;
  border: none;
  border-radius: 4px;
  cursor: pointer;
  font-size: 14px;
  transition: all 0.3s;
}
```

```
.btn-primary {
  background-color: #007bff;
  color: white;
}

.btn-primary:hover {
  background-color: #0056b3;
}

.btn-success {
  background-color: #28a745;
  color: white;
}

.btn-danger {
  background-color: #dc3545;
  color: white;
}

.form-group {
  margin-bottom: 15px;
}

.form-group label {
  display: block;
  margin-bottom: 5px;
  font-weight: 500;
}

.form-group input,
.form-group select,
.form-group textarea {
  width: 100%;
  padding: 10px;
  border: 1px solid #ddd;
  border-radius: 4px;
  font-size: 14px;
}

.form-group textarea {
  min-height: 100px;
  resize: vertical;
}

.alert {
  padding: 12px 20px;
  border-radius: 4px;
  margin-bottom: 20px;
}

.alert-success {
  background-color: #d4edda;
  color: #155724;
  border: 1px solid #c3e6cb;
}

.alert-error {
  background-color: #f8d7da;
  color: #721c24;
  border: 1px solid #f5c6cb;
}

.navbar {
  background-color: #343a40;
  color: white;
  padding: 15px 20px;
  display: flex;
  justify-content: space-between;
  align-items: center;
}
```

```
.navbar h1 {
  font-size: 24px;
}

.navbar-links {
  display: flex;
  gap: 20px;
}

.navbar-links a {
  color: white;
  text-decoration: none;
  padding: 8px 15px;
  border-radius: 4px;
  transition: background-color 0.3s;
}

.navbar-links a:hover {
  background-color: #495057;
}

.grid {
  display: grid;
  grid-template-columns: repeat(auto-fill, minmax(300px, 1fr));
  gap: 20px;
}

.loading {
  text-align: center;
  padding: 40px;
  font-size: 18px;
  color: #666;
}

.badge {
  display: inline-block;
  padding: 4px 8px;
  border-radius: 4px;
  font-size: 12px;
  font-weight: 500;
}

.badge-academic {
  background-color: #007bff;
  color: white;
}

.badge-club {
  background-color: #28a745;
  color: white;
}

.badge-pdf {
  background-color: #dc3545;
  color: white;
}

.badge-text {
  background-color: #6c757d;
  color: white;
}
```

5. src/App.jsx

```
import React from 'react';
import { BrowserRouter as Router, Routes, Route } from 'react-router-dom';
import { AuthProvider } from './context/AuthContext';
import ProtectedRoute from './components/ProtectedRoute';
import Navbar from './components/Layout/Navbar';
```

```
// Pages
import Login from './pages/Login';
import Register from './pages/Register';
import Dashboard from './pages/Dashboard';
import AcademicNotices from './pages/AcademicNotices';
import ClubActivities from './pages/ClubActivities';
import CreateNotice from './pages/CreateNotice';
import AdminPanel from './pages/AdminPanel';

function App() {
  return (
    &lt;Router&gt;
      &lt;AuthProvider&gt;
        <div>
          &lt;Navbar /&gt;
          &lt;Routes&gt;
            &lt;Route path="/" element={&lt;Login /&gt;} /&gt;
            &lt;Route path="/register" element={&lt;Register /&gt;} /&gt;

            &lt;Route element={&lt;ProtectedRoute /&gt;}&gt;
              &lt;Route path="/dashboard" element={&lt;Dashboard /&gt;} /&gt;
              &lt;Route path="/academic" element={&lt;AcademicNotices /&gt;} /&gt;
              &lt;Route path="/clubs" element={&lt;ClubActivities /&gt;} /&gt;
              &lt;Route path="/create-notice" element={&lt;CreateNotice /&gt;} /&gt;
              &lt;Route path="/admin" element={&lt;AdminPanel /&gt;} /&gt;
            &lt;/Route&gt;
          &lt;/Routes&gt;
        </div>
      &lt;/AuthProvider&gt;
    &lt;/Router&gt;
  );
}

export default App;
```

6. src/context/AuthContext.jsx

```
import React, { createContext, useState, useContext, useEffect } from 'react';
import api from '../services/api';

const AuthContext = createContext(null);

export const AuthProvider = ({ children }) =&gt; {
  const [user, setUser] = useState(null);
  const [loading, setLoading] = useState(true);
  const [token, setToken] = useState(localStorage.getItem('token') || null);

  useEffect(() =&gt; {
    // Check if user is logged in on mount
    const checkAuth = async () =&gt; {
      const storedToken = localStorage.getItem('token');
      if (storedToken) {
        try {
          const response = await api.get('/auth/me');
          setUser(response.data.user);
          setToken(storedToken);
        } catch (error) {
          console.error('Authentication check failed:', error);
          localStorage.removeItem('token');
          setToken(null);
        }
      }
      setLoading(false);
    };

    checkAuth();
  }, []);

  const login = async (email, password) =&gt; {
    try {
```

```
      const response = await api.post('/auth/login', { email, password });
      const { token, user } = response.data;

      localStorage.setItem('token', token);
      setToken(token);
      setUser(user);

      return { success: true };
    } catch (error) {
      return {
        success: false,
        message: error.response?.data?.message || 'Login failed',
      };
    }
  };

  const register = async (userData) =&gt; {
    try {
      const response = await api.post('/auth/register', userData);
      const { token, user } = response.data;

      localStorage.setItem('token', token);
      setToken(token);
      setUser(user);

      return { success: true };
    } catch (error) {
      return {
        success: false,
        message: error.response?.data?.message || 'Registration failed',
      };
    }
  };

  const logout = () =&gt; {
    localStorage.removeItem('token');
    setToken(null);
    setUser(null);
  };

  const value = {
    user,
    token,
    loading,
    login,
    register,
    logout,
    isAuthenticated: !!user,
  };

  return &lt;AuthContext.Provider value={value}&gt;{children}&lt;/AuthContext.Provider&gt;;
};

export const useAuth = () =&gt; {
  const context = useContext(AuthContext);
  if (!context) {
    throw new Error('useAuth must be used within an AuthProvider');
  }
  return context;
};
```

7. src/services/api.js

```
import axios from 'axios';

const api = axios.create({
  baseURL: process.env.REACT_APP_API_URL || 'http://localhost:5000/api',
  headers: {
    'Content-Type': 'application/json',
  },
```

```
});

// Add token to requests
api.interceptors.request.use(
  (config) =&gt; {
    const token = localStorage.getItem('token');
    if (token) {
      config.headers.Authorization = `Bearer ${token}`;
    }
    return config;
  },
  (error) =&gt; {
    return Promise.reject(error);
  }
);

// Handle response errors
api.interceptors.response.use(
  (response) =&gt; response,
  (error) =&gt; {
    if (error.response?.status === 401) {
      localStorage.removeItem('token');
      window.location.href = '/';
    }
    return Promise.reject(error);
  }
);

export default api;
```

8. src/components/ProtectedRoute.jsx

```
import React from 'react';
import { Navigate, Outlet } from 'react-router-dom';
import { useAuth } from '../context/AuthContext';

const ProtectedRoute = ({ allowedRoles }) =&gt; {
  const { user, loading } = useAuth();

  if (loading) {
    return <div>Loading...</div>;
  }

  if (!user) {
    return &lt;Navigate to="/" replace /&gt;;
  }

  if (allowedRoles &amp;&amp; !allowedRoles.includes(user.role)) {
    return &lt;Navigate to="/dashboard" replace /&gt;;
  }

  return &lt;Outlet /&gt;;
};

export default ProtectedRoute;
```

9. src/components/Layout/Navbar.jsx

```
import React from 'react';
import { Link, useNavigate } from 'react-router-dom';
import { useAuth } from '../../context/AuthContext';

const Navbar = () =&gt; {
  const { user, logout, isAuthenticated } = useAuth();
  const navigate = useNavigate();

  const handleLogout = () =&gt; {
    logout();
```

```
    navigate('/');
  };

  return (
    &lt;nav className="navbar"&gt;
      <h1>Student Activities</h1>
      {isAuthenticated &amp;&amp; (
        <div>
          &lt;Link to="/dashboard"&gt;Dashboard&lt;/Link&gt;
          &lt;Link to="/academic"&gt;Academic&lt;/Link&gt;
          &lt;Link to="/clubs"&gt;Clubs&lt;/Link&gt;
          {(user.canPost || user.role === 'admin') &amp;&amp; (
            &lt;Link to="/create-notice"&gt;Create Notice&lt;/Link&gt;
          )}
          {user.role === 'admin' &amp;&amp; (
            &lt;Link to="/admin"&gt;Admin Panel&lt;/Link&gt;
          )}
          <span>
            {user.name} ({user.role})
          </span>
          &lt;button
            className="btn btn-danger"
            onClick={handleLogout}
            style={{ marginLeft: '10px' }}
          &gt;
            Logout
          &lt;/button&gt;
        </div>
      )}
    &lt;/nav&gt;
  );
};

export default Navbar;
```

10. src/components/Notice/NoticeCard.jsx

```
import React from 'react';
import { FiCalendar, FiUser, FiFileText } from 'react-icons/fi';
import { BsFilePdf } from 'react-icons/bs';

const NoticeCard = ({ notice, onDelete, canDelete }) =&gt; {
  const formatDate = (date) =&gt; {
    return new Date(date).toLocaleDateString('en-US', {
      year: 'numeric',
      month: 'long',
      day: 'numeric',
    });
  };

  const handleViewPDF = () =&gt; {
    if (notice.pdfFile?.path) {
      window.open(
        `http://localhost:5000/${notice.pdfFile.path}`,
        '_blank'
      );
    }
  };

  return (
    <div>
      <div>
        <span>
          {notice.type.toUpperCase()}
        </span>
        <span>
          {notice.format.toUpperCase()}
        </span>
      </div>
```

```
      <h3>
        {notice.title}
      </h3>

      {notice.format === 'text' &amp;&amp; notice.content &amp;&amp; (
        <p>
          {notice.content.substring(0, 150)}
          {notice.content.length &gt; 150 &amp;&amp; '...'}
        </p>
      )}

      {notice.format === 'pdf' &amp;&amp; notice.pdfFile &amp;&amp; (
        &lt;button
          className="btn btn-primary"
          onClick={handleViewPDF}
          style={{ marginBottom: '15px', display: 'flex', alignItems: 'center', gap: '5px' }}
        &gt;
          &lt;BsFilePdf /&gt; View PDF
        &lt;/button&gt;
      )}

      <div>
        <div>
          &lt;FiUser /&gt;
          <span>Posted by: {notice.postedBy?.name || 'Unknown'}</span>
        </div>
        <div>
          &lt;FiCalendar /&gt;
          <span>{formatDate(notice.createdAt)}</span>
        </div>
        {notice.department &amp;&amp; (
          <div>
            &lt;FiFileText /&gt;
            <span>Department: {notice.department.name}</span>
          </div>
        )}
        {notice.club &amp;&amp; (
          <div>
            &lt;FiFileText /&gt;
            <span>Club: {notice.club.name}</span>
          </div>
        )}
      </div>

      {canDelete &amp;&amp; (
        &lt;button
          className="btn btn-danger"
          onClick={() =&gt; onDelete(notice._id)}
          style={{ marginTop: '15px' }}
        &gt;
          Delete
        &lt;/button&gt;
      )}
    </div>
  );
};

export default NoticeCard;
```

11. src/components/Notice/NoticeForm.jsx

```
import React, { useState, useEffect } from 'react';
import api from '../../services/api';

const NoticeForm = ({ onSuccess }) =&gt; {
  const [formData, setFormData] = useState({
    title: '',
    content: '',
    type: 'academic',
    format: 'text',
```

```
  department: '',
  club: '',
});
const [pdfFile, setPdfFile] = useState(null);
const [departments, setDepartments] = useState([]);
const [clubs, setClubs] = useState([]);
const [loading, setLoading] = useState(false);
const [error, setError] = useState('');
const [success, setSuccess] = useState('');

useEffect(() =&gt; {
  fetchDepartments();
  fetchClubs();
}, []);

const fetchDepartments = async () =&gt; {
  try {
    const response = await api.get('/departments');
    setDepartments(response.data.departments);
  } catch (error) {
    console.error('Error fetching departments:', error);
  }
};

const fetchClubs = async () =&gt; {
  try {
    const response = await api.get('/clubs');
    setClubs(response.data.clubs);
  } catch (error) {
    console.error('Error fetching clubs:', error);
  }
};

const handleChange = (e) =&gt; {
  setFormData({
    ...formData,
    [e.target.name]: e.target.value,
  });
};

const handleFileChange = (e) =&gt; {
  setPdfFile(e.target.files[0]);
};

const handleSubmit = async (e) =&gt; {
  e.preventDefault();
  setLoading(true);
  setError('');
  setSuccess('');

  try {
    const submitData = new FormData();
    submitData.append('title', formData.title);
    submitData.append('type', formData.type);
    submitData.append('format', formData.format);

    if (formData.format === 'text') {
      submitData.append('content', formData.content);
    } else if (formData.format === 'pdf' &amp;&amp; pdfFile) {
      submitData.append('pdfFile', pdfFile);
    }

    if (formData.type === 'academic' &amp;&amp; formData.department) {
      submitData.append('department', formData.department);
    } else if (formData.type === 'club' &amp;&amp; formData.club) {
      submitData.append('club', formData.club);
    }

    await api.post('/notices', submitData, {
      headers: {
        'Content-Type': 'multipart/form-data',
      },
```

```
      });

      setSuccess('Notice created successfully!');
      setFormData({
        title: '',
        content: '',
        type: 'academic',
        format: 'text',
        department: '',
        club: '',
      });
      setPdfFile(null);

      if (onSuccess) {
        onSuccess();
      }
    } catch (error) {
      setError(error.response?.data?.message || 'Failed to create notice');
    } finally {
      setLoading(false);
    }
  };

  return (
    <div>
      <h2>Create New Notice</h2>
      {error &amp;&amp; <div>{error}</div>}
      {success &amp;&amp; <div>{success}</div>}

      &lt;form onSubmit={handleSubmit}&gt;
        <div>
          &lt;label&gt;Title *&lt;/label&gt;
          &lt;input
            type="text"
            name="title"
            value={formData.title}
            onChange={handleChange}
            required
          /&gt;
        </div>

        <div>
          &lt;label&gt;Type *&lt;/label&gt;
          &lt;select name="type" value={formData.type} onChange={handleChange} required&gt;
            &lt;option value="academic"&gt;Academic&lt;/option&gt;
            &lt;option value="club"&gt;Club&lt;/option&gt;
          &lt;/select&gt;
        </div>

        <div>
          &lt;label&gt;Format *&lt;/label&gt;
          &lt;select name="format" value={formData.format} onChange={handleChange} required&gt;
            &lt;option value="text"&gt;Text&lt;/option&gt;
            &lt;option value="pdf"&gt;PDF&lt;/option&gt;
          &lt;/select&gt;
        </div>

        {formData.type === 'academic' &amp;&amp; (
          <div>
            &lt;label&gt;Department&lt;/label&gt;
            &lt;select name="department" value={formData.department} onChange={handleChange}&gt;
              &lt;option value=""&gt;Select Department&lt;/option&gt;
              {departments.map((dept) =&gt; (
                &lt;option key={dept._id} value={dept._id}&gt;
                  {dept.name}
                &lt;/option&gt;
              ))}
            &lt;/select&gt;
          </div>
        )}

        {formData.type === 'club' &amp;&amp; (
```

```
          <div>
            &lt;label&gt;Club&lt;/label&gt;
            &lt;select name="club" value={formData.club} onChange={handleChange}&gt;
              &lt;option value=""&gt;Select Club&lt;/option&gt;
              {clubs.map((club) =&gt; (
                &lt;option key={club._id} value={club._id}&gt;
                  {club.name}
                &lt;/option&gt;
              ))}
            &lt;/select&gt;
          </div>
        )}

        {formData.format === 'text' &amp;&amp; (
          <div>
            &lt;label&gt;Content *&lt;/label&gt;
            &lt;textarea
              name="content"
              value={formData.content}
              onChange={handleChange}
              required={formData.format === 'text'}
            /&gt;
          </div>
        )}

        {formData.format === 'pdf' &amp;&amp; (
          <div>
            &lt;label&gt;Upload PDF *&lt;/label&gt;
            &lt;input
              type="file"
              accept=".pdf"
              onChange={handleFileChange}
              required={formData.format === 'pdf'}
            /&gt;
          </div>
        )}

        &lt;button type="submit" className="btn btn-primary" disabled={loading}&gt;
          {loading ? 'Creating...' : 'Create Notice'}
        &lt;/button&gt;
      &lt;/form&gt;
    </div>
  );
};

export default NoticeForm;
```

12. src/components/Notice/NoticeList.jsx

```
import React, { useState, useEffect } from 'react';
import api from '../../services/api';
import NoticeCard from './NoticeCard';
import { useAuth } from '../../context/AuthContext';

const NoticeList = ({ type, department, club }) =&gt; {
  const [notices, setNotices] = useState([]);
  const [loading, setLoading] = useState(true);
  const { user } = useAuth();

  useEffect(() =&gt; {
    fetchNotices();
  }, [type, department, club]);

  const fetchNotices = async () =&gt; {
    try {
      let url = '/notices?';
      if (type) url += `type=${type}&amp;`;
      if (department) url += `department=${department}&amp;`;
      if (club) url += `club=${club}&amp;`;
```

```
      const response = await api.get(url);
      setNotices(response.data.notices);
    } catch (error) {
      console.error('Error fetching notices:', error);
    } finally {
      setLoading(false);
    }
  };

  const handleDelete = async (noticeId) =&gt; {
    if (window.confirm('Are you sure you want to delete this notice?')) {
      try {
        await api.delete(`/notices/${noticeId}`);
        fetchNotices();
      } catch (error) {
        alert('Failed to delete notice');
      }
    }
  };

  const canDeleteNotice = (notice) =&gt; {
    return (
      user.role === 'admin' ||
      notice.postedBy?._id === user.id
    );
  };

  if (loading) {
    return <div>Loading notices...</div>;
  }

  if (notices.length === 0) {
    return (
      <div>
        <p>No notices available.</p>
      </div>
    );
  }

  return (
    <div>
      {notices.map((notice) =&gt; (
        &lt;NoticeCard
          key={notice._id}
          notice={notice}
          onDelete={handleDelete}
          canDelete={canDeleteNotice(notice)}
        /&gt;
      ))}
    </div>
  );
};

export default NoticeList;
```

13. src/components/Admin/UserManagement.jsx

```
import React, { useState, useEffect } from 'react';
import api from '../../services/api';

const UserManagement = () =&gt; {
  const [users, setUsers] = useState([]);
  const [loading, setLoading] = useState(true);

  useEffect(() =&gt; {
    fetchUsers();
  }, []);

  const fetchUsers = async () =&gt; {
    try {
```

```
    const response = await api.get('/users');
    setUsers(response.data.users);
  } catch (error) {
    console.error('Error fetching users:', error);
  } finally {
    setLoading(false);
  }
};

const togglePostingPrivilege = async (userId, currentStatus) =&gt; {
  try {
    await api.put(`/users/${userId}/privileges`, {
      canPost: !currentStatus,
    });
    fetchUsers();
  } catch (error) {
    alert('Failed to update privileges');
  }
};

const updateRole = async (userId, newRole) =&gt; {
  try {
    await api.put(`/users/${userId}/role`, { role: newRole });
    fetchUsers();
  } catch (error) {
    alert('Failed to update role');
  }
};

if (loading) {
  return <div>Loading users...</div>;
}

return (
  <div>
    <h2>User Management</h2>
    <div>
      &lt;table style={{ width: '100%', borderCollapse: 'collapse' }}&gt;
        &lt;thead&gt;
          &lt;tr style={{ borderBottom: '2px solid #dee2e6' }}&gt;
            &lt;th style={{ padding: '12px', textAlign: 'left' }}&gt;Name&lt;/th&gt;
            &lt;th style={{ padding: '12px', textAlign: 'left' }}&gt;Email&lt;/th&gt;
            &lt;th style={{ padding: '12px', textAlign: 'left' }}&gt;Role&lt;/th&gt;
            &lt;th style={{ padding: '12px', textAlign: 'left' }}&gt;Can Post&lt;/th&gt;
            &lt;th style={{ padding: '12px', textAlign: 'left' }}&gt;Actions&lt;/th&gt;
          &lt;/tr&gt;
        &lt;/thead&gt;
        &lt;tbody&gt;
          {users.map((user) =&gt; (
            &lt;tr key={user._id} style={{ borderBottom: '1px solid #dee2e6' }}&gt;
              &lt;td style={{ padding: '12px' }}&gt;{user.name}&lt;/td&gt;
              &lt;td style={{ padding: '12px' }}&gt;{user.email}&lt;/td&gt;
              &lt;td style={{ padding: '12px' }}&gt;
                &lt;select
                  value={user.role}
                  onChange={(e) =&gt; updateRole(user._id, e.target.value)}
                  style={{ padding: '5px' }}
                &gt;
                  &lt;option value="student"&gt;Student&lt;/option&gt;
                  &lt;option value="faculty"&gt;Faculty&lt;/option&gt;
                  &lt;option value="club_member"&gt;Club Member&lt;/option&gt;
                  &lt;option value="admin"&gt;Admin&lt;/option&gt;
                &lt;/select&gt;
              &lt;/td&gt;
              &lt;td style={{ padding: '12px' }}&gt;
                &lt;input
                  type="checkbox"
                  checked={user.canPost}
                  onChange={() =&gt; togglePostingPrivilege(user._id, user.canPost)}
                /&gt;
              &lt;/td&gt;
              &lt;td style={{ padding: '12px' }}&gt;
```

```
                  <span>
                    {user.canPost ? 'Can Post' : 'Cannot Post'}
                  </span>
                &lt;/td&gt;
              &lt;/tr&gt;
            ))}
          &lt;/tbody&gt;
        &lt;/table&gt;
      </div>
    </div>
  );
};

export default UserManagement;
```

14. src/pages/Login.jsx

```
import React, { useState } from 'react';
import { useNavigate, Link } from 'react-router-dom';
import { useAuth } from '../context/AuthContext';

const Login = () =&gt; {
  const [formData, setFormData] = useState({
    email: '',
    password: '',
  });
  const [error, setError] = useState('');
  const [loading, setLoading] = useState(false);
  const navigate = useNavigate();
  const { login } = useAuth();

  const handleChange = (e) =&gt; {
    setFormData({
      ...formData,
      [e.target.name]: e.target.value,
    });
  };

  const handleSubmit = async (e) =&gt; {
    e.preventDefault();
    setLoading(true);
    setError('');

    const result = await login(formData.email, formData.password);

    if (result.success) {
      navigate('/dashboard');
    } else {
      setError(result.message);
    }

    setLoading(false);
  };

  return (
    <div>
      <div>
        <h2>
          Login to Student Activities
        </h2>

        {error &amp;&amp; <div>{error}</div>}

        &lt;form onSubmit={handleSubmit}&gt;
          <div>
            &lt;label&gt;Email&lt;/label&gt;
            &lt;input
              type="email"
              name="email"
              value={formData.email}
```

```
              onChange={handleChange}
              required
            /&gt;
          </div>

          <div>
            &lt;label&gt;Password&lt;/label&gt;
            &lt;input
              type="password"
              name="password"
              value={formData.password}
              onChange={handleChange}
              required
            /&gt;
          </div>

          &lt;button
            type="submit"
            className="btn btn-primary"
            style={{ width: '100%' }}
            disabled={loading}
          &gt;
            {loading ? 'Logging in...' : 'Login'}
          &lt;/button&gt;
        &lt;/form&gt;

        <p>
          Don't have an account? &lt;Link to="/register"&gt;Register here&lt;/Link&gt;
        </p>
      </div>
    </div>
  );
};

export default Login;
```

15. src/pages/Register.jsx

```
import React, { useState, useEffect } from 'react';
import { useNavigate, Link } from 'react-router-dom';
import { useAuth } from '../context/AuthContext';
import api from '../services/api';

const Register = () =&gt; {
  const [formData, setFormData] = useState({
    name: '',
    email: '',
    password: '',
    role: 'student',
    department: '',
    club: '',
  });
  const [departments, setDepartments] = useState([]);
  const [clubs, setClubs] = useState([]);
  const [error, setError] = useState('');
  const [loading, setLoading] = useState(false);
  const navigate = useNavigate();
  const { register } = useAuth();

  useEffect(() =&gt; {
    fetchDepartments();
    fetchClubs();
  }, []);

  const fetchDepartments = async () =&gt; {
    try {
      const response = await api.get('/departments');
      setDepartments(response.data.departments);
    } catch (error) {
      console.error('Error fetching departments:', error);
```

```
  }
};

const fetchClubs = async () =&gt; {
  try {
    const response = await api.get('/clubs');
    setClubs(response.data.clubs);
  } catch (error) {
    console.error('Error fetching clubs:', error);
  }
};

const handleChange = (e) =&gt; {
  setFormData({
    ...formData,
    [e.target.name]: e.target.value,
  });
};

const handleSubmit = async (e) =&gt; {
  e.preventDefault();
  setLoading(true);
  setError('');

  const result = await register(formData);

  if (result.success) {
    navigate('/dashboard');
  } else {
    setError(result.message);
  }

  setLoading(false);
};

return (
  <div>
    <div>
      <h2>
        Register for Student Activities
      </h2>

      {error &amp;&amp; <div>{error}</div>}

      &lt;form onSubmit={handleSubmit}&gt;
        <div>
          &lt;label&gt;Name&lt;/label&gt;
          &lt;input
            type="text"
            name="name"
            value={formData.name}
            onChange={handleChange}
            required
          /&gt;
        </div>

        <div>
          &lt;label&gt;Email&lt;/label&gt;
          &lt;input
            type="email"
            name="email"
            value={formData.email}
            onChange={handleChange}
            required
          /&gt;
        </div>

        <div>
          &lt;label&gt;Password&lt;/label&gt;
          &lt;input
            type="password"
            name="password"
```

```
              value={formData.password}
              onChange={handleChange}
              required
              minLength={6}
            /&gt;
          </div>

          <div>
            &lt;label&gt;Role&lt;/label&gt;
            &lt;select name="role" value={formData.role} onChange={handleChange} required&gt;
              &lt;option value="student"&gt;Student&lt;/option&gt;
              &lt;option value="faculty"&gt;Faculty&lt;/option&gt;
              &lt;option value="club_member"&gt;Club Member&lt;/option&gt;
            &lt;/select&gt;
          </div>

          <div>
            &lt;label&gt;Department&lt;/label&gt;
            &lt;select name="department" value={formData.department} onChange={handleChange}&gt;
              &lt;option value=""&gt;Select Department (Optional)&lt;/option&gt;
              {departments.map((dept) =&gt; (
                &lt;option key={dept._id} value={dept._id}&gt;
                  {dept.name}
                &lt;/option&gt;
              ))}
            &lt;/select&gt;
          </div>

          <div>
            &lt;label&gt;Club&lt;/label&gt;
            &lt;select name="club" value={formData.club} onChange={handleChange}&gt;
              &lt;option value=""&gt;Select Club (Optional)&lt;/option&gt;
              {clubs.map((club) =&gt; (
                &lt;option key={club._id} value={club._id}&gt;
                  {club.name}
                &lt;/option&gt;
              ))}
            &lt;/select&gt;
          </div>

          &lt;button
            type="submit"
            className="btn btn-primary"
            style={{ width: '100%' }}
            disabled={loading}
          &gt;
            {loading ? 'Registering...' : 'Register'}
          &lt;/button&gt;
        &lt;/form&gt;

        <p>
          Already have an account? &lt;Link to="/"&gt;Login here&lt;/Link&gt;
        </p>
      </div>
    </div>
  );
};

export default Register;
```

16. src/pages/Dashboard.jsx

```
import React from 'react';
import { useAuth } from '../context/AuthContext';
import NoticeList from '../components/Notice/NoticeList';

const Dashboard = () =&gt; {
  const { user } = useAuth();

  return (
```

```
    <div>
      <div>
        <h1>Welcome, {user.name}!</h1>
        <p>Role: <strong>{user.role}</strong></p>
        {user.department &amp;&amp; (
          <p>Department: <strong>{user.department.name}</strong></p>
        )}
        {user.club &amp;&amp; (
          <p>Club: <strong>{user.club.name}</strong></p>
        )}
        <p>
          Posting Privileges:{' '}
          <strong>{user.canPost || user.role === 'admin' ? 'Enabled' : 'Disabled'}</strong>
        </p>
      </div>

      <h2>Recent Notices</h2>
      &lt;NoticeList /&gt;
    </div>
  );
};

export default Dashboard;
```

17. src/pages/AcademicNotices.jsx

```
import React from 'react';
import NoticeList from '../components/Notice/NoticeList';

const AcademicNotices = () =&gt; {
  return (
    <div>
      <h1>Academic Notices</h1>
      &lt;NoticeList type="academic" /&gt;
    </div>
  );
};

export default AcademicNotices;
```

18. src/pages/ClubActivities.jsx

```
import React from 'react';
import NoticeList from '../components/Notice/NoticeList';

const ClubActivities = () =&gt; {
  return (
    <div>
      <h1>Club Activities</h1>
      &lt;NoticeList type="club" /&gt;
    </div>
  );
};

export default ClubActivities;
```

19. src/pages/CreateNotice.jsx

```
import React from 'react';
import { useNavigate } from 'react-router-dom';
import NoticeForm from '../components/Notice/NoticeForm';

const CreateNotice = () =&gt; {
  const navigate = useNavigate();

  const handleSuccess = () =&gt; {
    setTimeout(() =&gt; {
```

```
      navigate('/dashboard');
    }, 2000);
  };

  return (
    <div>
      <h1>Create New Notice</h1>
      &lt;NoticeForm onSuccess={handleSuccess} /&gt;
    </div>
  );
};

export default CreateNotice;
```

20. src/pages/AdminPanel.jsx

```
import React from 'react';
import { useAuth } from '../context/AuthContext';
import { Navigate } from 'react-router-dom';
import UserManagement from '../components/Admin/UserManagement';

const AdminPanel = () =&gt; {
  const { user } = useAuth();

  if (user.role !== 'admin') {
    return &lt;Navigate to="/dashboard" replace /&gt;;
  }

  return (
    <div>
      <h1>Admin Panel</h1>
      &lt;UserManagement /&gt;
    </div>
  );
};

export default AdminPanel;
```

Installation Instructions

1. Create React App

```
npx create-react-app frontend
cd frontend
```

2. Install Dependencies

```
npm install react-router-dom axios react-icons
```

3. Replace src folder

Replace the contents of the `src` folder with the files provided above.

4. Configure Environment

Create a `.env` file in the frontend root directory with your API URL.

5. Run the Application

```
npm start
```

The frontend will run on `http://localhost:3000`

Features Summary

Authentication

- Login and Registration with JWT
- Protected routes
- Role-based access control

Notice Management

- View all notices (Academic/Club)
- Create text or PDF notices
- Filter by department or club
- Delete own notices (or all if admin)

Admin Features

- Manage user roles
- Grant/revoke posting privileges
- View all users

User Roles

- **Student**: View notices only
- **Faculty**: Can post academic notices (if granted)
- **Club Member**: Can post club notices (if granted)
- **Admin**: Full access to all features