Creating a complete, fully functional React app is a complex task. However, I can provide the foundational React components, their corresponding CSS files, and a simple project structure that you can use as a starting point. This setup will give you the core dashboard, task, and note management components.

**Project Structure**

First, here's how you can organize the files in your React project. You would typically create this structure using a tool like Vite or create-react-app.

/src

├── App.jsx

├── index.css

├── main.jsx

├── components/

│ ├── Dashboard/

│ │ ├── Dashboard.jsx

│ │ └── Dashboard.css

│ ├── Sidebar/

│ │ ├── Sidebar.jsx

│ │ └── Sidebar.css

│ ├── MyNotes/

│ │ ├── MyNotes.jsx

│ │ └── MyNotes.css

│ ├── Tasks/

│ │ ├── Tasks.jsx

│ │ └── Tasks.css

**App.jsx**

This is the main component that renders the entire application layout, including the Sidebar and the main content area. We'll use a simple state to manage which component is currently displayed.

JavaScript

import React, { useState } from 'react';

import Sidebar from './components/Sidebar/Sidebar';

import Dashboard from './components/Dashboard/Dashboard';

import MyNotes from './components/MyNotes/MyNotes';

import Tasks from './components/Tasks/Tasks';

function App() {

const [activeSection, setActiveSection] = useState('dashboard');

const renderSection = () => {

switch (activeSection) {

case 'dashboard':

return <Dashboard />;

case 'my-notes':

return <MyNotes />;

case 'tasks':

return <Tasks />;

default:

return <Dashboard />;

}

};

return (

<div className="app-container">

<Sidebar setActiveSection={setActiveSection} />

<div className="main-content">

{renderSection()}

</div>

</div>

);

}

export default App;

**src/index.css**

This file contains the global styles for your application, including the main layout for the App.jsx container.

CSS

body {

margin: 0;

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: #f0f2f5;

color: #333;

}

.app-container {

display: flex;

min-height: 100vh;

}

.main-content {

flex-grow: 1;

padding: 2rem;

}

**components/Sidebar/Sidebar.jsx**

The Sidebar component handles navigation and quick actions.

JavaScript

import React from 'react';

import './Sidebar.css';

const Sidebar = ({ setActiveSection }) => {

return (

<div className="sidebar">

<div className="logo-container">

<h3>Agent Compass</h3>

</div>

<nav className="nav-menu">

<ul>

<li onClick={() => setActiveSection('dashboard')}>Dashboard</li>

<li onClick={() => setActiveSection('my-notes')}>My Notes</li>

<li onClick={() => setActiveSection('tasks')}>Tasks</li>

</ul>

</nav>

<div className="quick-actions">

<h4>Quick Actions</h4>

<button className="action-btn">Upload Notes</button>

<button className="action-btn">Add New Task</button>

<button className="action-btn">Ask Agent</button>

</div>

</div>

);

};

export default Sidebar;

**components/Sidebar/Sidebar.css**

CSS

.sidebar {

width: 250px;

background-color: #fff;

padding: 2rem 1.5rem;

box-shadow: 2px 0 5px rgba(0, 0, 0, 0.05);

display: flex;

flex-direction: column;

}

.logo-container {

margin-bottom: 2rem;

text-align: center;

}

.nav-menu ul {

list-style: none;

padding: 0;

margin: 0;

}

.nav-menu li {

padding: 0.8rem 1rem;

margin-bottom: 0.5rem;

cursor: pointer;

border-radius: 8px;

transition: background-color 0.2s ease-in-out;

}

.nav-menu li:hover {

background-color: #e9ecef;

}

.quick-actions {

margin-top: 2rem;

}

.quick-actions h4 {

margin-bottom: 1rem;

}

.action-btn {

width: 100%;

padding: 0.75rem;

margin-bottom: 0.75rem;

border: 1px solid #ced4da;

border-radius: 8px;

background-color: #fff;

cursor: pointer;

transition: all 0.2s ease-in-out;

}

.action-btn:hover {

background-color: #e9ecef;

border-color: #adb5bd;

}

**components/Dashboard/Dashboard.jsx**

This is the main dashboard view, consisting of various cards.

JavaScript

import React from 'react';

import './Dashboard.css';

const Dashboard = () => {

return (

<div className="dashboard">

<h2>Welcome, Student! 👋</h2>

<div className="dashboard-grid">

<div className="card upcoming-tasks">

<h3>Upcoming Deadlines</h3>

<ul>

<li>📝 Essay - Modern History (Due in 3 days)</li>

<li>📊 Project Proposal - CS (Due in 7 days)</li>

</ul>

</div>

<div className="card recent-summaries">

<h3>Recent Summaries</h3>

<ul>

<li><a href="#">Quantum Mechanics Lecture 1</a></li>

<li><a href="#">The Industrial Revolution</a></li>

</ul>

</div>

<div className="card recommendations">

<h3>Personalized Recommendations</h3>

<p>Based on your notes on physics, you may find these helpful:</p>

<ul>

<li>Video: Feynman Lectures on Physics</li>

<li>Practice: Problem Set 2 on Khan Academy</li>

</ul>

</div>

</div>

</div>

);

};

export default Dashboard;

**components/Dashboard/Dashboard.css**

CSS

.dashboard {

flex-grow: 1;

}

.dashboard-grid {

display: grid;

grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));

gap: 2rem;

margin-top: 2rem;

}

.card {

background-color: #fff;

padding: 1.5rem;

border-radius: 12px;

box-shadow: 0 4px 6px rgba(0, 0, 0, 0.05);

}

.card h3 {

margin-top: 0;

color: #555;

}

.card ul {

list-style: none;

padding: 0;

}

.card li {

margin-bottom: 0.75rem;

}

**components/MyNotes/MyNotes.jsx and Tasks.jsx**

These files follow a similar pattern. They'll have a header, an input area (e.g., file upload), and a list of items.

**components/MyNotes/MyNotes.jsx**

JavaScript

import React from 'react';

import './MyNotes.css';

const MyNotes = () => {

return (

<div className="my-notes">

<h2>My Notes</h2>

<div className="upload-section">

<label htmlFor="file-upload" className="upload-btn">

Upload New Document

</label>

<input id="file-upload" type="file" style={{ display: 'none' }} />

<p>Drag & drop files here</p>

</div>

<div className="notes-list">

<h4>Uploaded Documents</h4>

<ul>

<li>Quantum Mechanics Lecture 1.pdf (Uploaded: Oct 26, 2024)</li>

<li>The Industrial Revolution.pdf (Uploaded: Oct 25, 2024)</li>

</ul>

</div>

</div>

);

};

export default MyNotes;

**components/Tasks/Tasks.jsx**

JavaScript

import React from 'react';

import './Tasks.css';

const Tasks = () => {

return (

<div className="tasks">

<h2>Tasks & Deadlines</h2>

<div className="add-task-section">

<input type="text" placeholder="Enter new task..." />

<button>Add Task</button>

</div>

<div className="task-list">

<h4>My To-Do List</h4>

<ul>

<li><input type="checkbox" /> Write essay on Modern History</li>

<li><input type="checkbox" /> Prepare CS project proposal</li>

</ul>

</div>

</div>

);

};

export default Tasks;

**Note:** You would need to create the corresponding .css files for MyNotes.jsx and Tasks.jsx with similar styling to the other components (e.g., using .my-notes and .tasks selectors to style the content within those components). The code provided above gives you the core React structure and logic to build upon.

give me mynotes and task files

Show thinking

Okay, here are the React files for the MyNotes and Tasks components, including the JSX and CSS, based on the front-end preview we discussed. These components are designed to be simple and functional, providing a solid foundation for you to build upon.

**components/MyNotes/MyNotes.jsx**

This component handles the display of uploaded notes and provides a user-friendly interface for uploading new files using a hidden <input type="file"> element.

JavaScript

import React, { useState } from 'react';

import './MyNotes.css';

const MyNotes = () => {

const [uploadedFiles, setUploadedFiles] = useState([

{ name: 'Quantum Mechanics Lecture 1.pdf', date: 'Oct 26, 2025' },

{ name: 'The Industrial Revolution.pdf', date: 'Oct 25, 2025' },

{ name: 'Machine Learning Basics.pdf', date: 'Oct 24, 2025' },

]);

const handleFileUpload = (event) => {

// Get the files from the input element

const files = event.target.files;

// A simple check to ensure files were selected

if (files.length > 0) {

// For this example, we'll just add the file names to our state

const newFiles = Array.from(files).map(file => ({

name: file.name,

date: new Date().toLocaleDateString('en-US', {

month: 'short',

day: '2-digit',

year: 'numeric',

}),

}));

setUploadedFiles(prevFiles => [...newFiles, ...prevFiles]);

// You would add logic here to send the file to your backend server

console.log('Files selected for upload:', files);

}

};

return (

<div className="my-notes-container">

<h2>My Notes</h2>

<div className="upload-box">

<label htmlFor="file-upload" className="upload-btn">

Upload New Document

</label>

<input

id="file-upload"

type="file"

multiple

onChange={handleFileUpload}

style={{ display: 'none' }}

/>

<p className="drag-and-drop-text">or drag & drop files here</p>

</div>

<div className="notes-list">

<h4>Uploaded Documents</h4>

{uploadedFiles.length > 0 ? (

<ul>

{uploadedFiles.map((file, index) => (

<li key={index}>

<span className="file-name">📄 {file.name}</span>

<span className="upload-date">Uploaded: {file.date}</span>

<button className="view-summary-btn">View Summary</button>

</li>

))}

</ul>

) : (

<p className="empty-state">You haven't uploaded any notes yet.</p>

)}

</div>

</div>

);

};

export default MyNotes;

**components/MyNotes/MyNotes.css**

CSS

.my-notes-container {

padding: 2rem;

}

.my-notes-container h2 {

margin-bottom: 1.5rem;

color: #333;

}

.upload-box {

border: 2px dashed #ccc;

border-radius: 12px;

padding: 2rem;

text-align: center;

background-color: #f9f9f9;

transition: border-color 0.2s ease-in-out;

margin-bottom: 2rem;

}

.upload-box:hover {

border-color: #007bff;

}

.upload-btn {

background-color: #007bff;

color: white;

padding: 0.75rem 1.5rem;

border-radius: 8px;

cursor: pointer;

font-weight: bold;

transition: background-color 0.2s ease-in-out;

}

.upload-btn:hover {

background-color: #0056b3;

}

.drag-and-drop-text {

margin-top: 1rem;

color: #888;

font-style: italic;

}

.notes-list {

background-color: #fff;

padding: 1.5rem;

border-radius: 12px;

box-shadow: 0 4px 6px rgba(0, 0, 0, 0.05);

}

.notes-list h4 {

margin-top: 0;

margin-bottom: 1rem;

color: #555;

border-bottom: 1px solid #eee;

padding-bottom: 0.5rem;

}

.notes-list ul {

list-style: none;

padding: 0;

margin: 0;

}

.notes-list li {

display: flex;

justify-content: space-between;

align-items: center;

padding: 0.75rem 0;

border-bottom: 1px solid #f0f2f5;

}

.notes-list li:last-child {

border-bottom: none;

}

.file-name {

font-weight: 500;

}

.upload-date {

font-size: 0.9em;

color: #888;

}

.view-summary-btn {

background: none;

border: 1px solid #ced4da;

padding: 0.5rem 1rem;

border-radius: 8px;

cursor: pointer;

transition: all 0.2s ease-in-out;

}

.view-summary-btn:hover {

background-color: #f0f2f5;

color: #007bff;

}

.empty-state {

text-align: center;

color: #888;

padding: 2rem;

}

**components/Tasks/Tasks.jsx**

This component manages the student's task list, allowing for new task creation and tracking with a simple checkbox.

JavaScript

import React, { useState } from 'react';

import './Tasks.css';

const Tasks = () => {

const [tasks, setTasks] = useState([

{ id: 1, text: 'Write essay on Modern History', completed: false, dueDate: 'Oct 30, 2025' },

{ id: 2, text: 'Prepare CS project proposal', completed: false, dueDate: 'Nov 7, 2025' },

{ id: 3, text: 'Read Chapter 5 for Biology class', completed: true, dueDate: 'Oct 22, 2025' },

]);

const [newTask, setNewTask] = useState('');

const handleAddTask = () => {

if (newTask.trim() !== '') {

const newId = tasks.length > 0 ? Math.max(...tasks.map(t => t.id)) + 1 : 1;

const newDueDate = new Date().toLocaleDateString('en-US', {

month: 'short',

day: '2-digit',

year: 'numeric',

});

setTasks([...tasks, { id: newId, text: newTask, completed: false, dueDate: newDueDate }]);

setNewTask('');

}

};

const handleToggleComplete = (id) => {

setTasks(tasks.map(task =>

task.id === id ? { ...task, completed: !task.completed } : task

));

};

return (

<div className="tasks-container">

<h2>Tasks & Deadlines</h2>

<div className="add-task-section">

<input

type="text"

placeholder="Enter a new task..."

value={newTask}

onChange={(e) => setNewTask(e.target.value)}

onKeyPress={(e) => {

if (e.key === 'Enter') handleAddTask();

}}

/>

<button onClick={handleAddTask}>Add Task</button>

</div>

<div className="task-list">

{tasks.length > 0 ? (

<ul>

{tasks.map(task => (

<li key={task.id} className={task.completed ? 'completed' : ''}>

<label className="task-item">

<input

type="checkbox"

checked={task.completed}

onChange={() => handleToggleComplete(task.id)}

/>

<span className="task-text">{task.text}</span>

<span className="due-date">Due: {task.dueDate}</span>

</label>

</li>

))}

</ul>

) : (

<p className="empty-state">No tasks to display! Great job!</p>

)}

</div>

</div>

);

};

export default Tasks;

**components/Tasks/Tasks.css**

CSS

.tasks-container {

padding: 2rem;

}

.tasks-container h2 {

margin-bottom: 1.5rem;

color: #333;

}

.add-task-section {

display: flex;

gap: 1rem;

margin-bottom: 2rem;

}

.add-task-section input {

flex-grow: 1;

padding: 0.75rem;

border: 1px solid #ccc;

border-radius: 8px;

font-size: 1rem;

}

.add-task-section button {

padding: 0.75rem 1.5rem;

background-color: #007bff;

color: white;

border: none;

border-radius: 8px;

cursor: pointer;

transition: background-color 0.2s ease-in-out;

}

.add-task-section button:hover {

background-color: #0056b3;

}

.task-list {

background-color: #fff;

padding: 1.5rem;

border-radius: 12px;

box-shadow: 0 4px 6px rgba(0, 0, 0, 0.05);

}

.task-list h4 {

margin-top: 0;

margin-bottom: 1rem;

color: #555;

border-bottom: 1px solid #eee;

padding-bottom: 0.5rem;

}

.task-list ul {

list-style: none;

padding: 0;

margin: 0;

}

.task-list li {

padding: 0.75rem 0;

border-bottom: 1px solid #f0f2f5;

}

.task-list li:last-child {

border-bottom: none;

}

.task-item {

display: flex;

align-items: center;

cursor: pointer;

position: relative;

}

.task-item input[type="checkbox"] {

margin-right: 1rem;

}

.task-text {

flex-grow: 1;

font-weight: 500;

}

.due-date {

font-size: 0.9em;

color: #888;

margin-left: 1rem;

}

.completed .task-text, .completed .due-date {

text-decoration: line-through;

color: #aaa;

}

.empty-state {

text-align: center;

color: #888;

padding: 2rem;

}