**Tender Insight Hub - Project Leadership Report**

**Course: NSED742 - Software Engineering & Design  
Project: Tender Insight Hub - SaaS Platform for SME Procurement  
Team Lead: Khethiwe Skosana  
Date: October 2025**

## 1. \*Phase Overview\*

**\*Objective:\***

**Integrate the FastAPI backend (Phase 2–3) with an interactive frontend interface that allows users to search tenders, view AI summaries, check readiness scores, and manage their workspace within a unified SaaS dashboard.**

**\*Tools & Frameworks\***

**\* \*Frontend:\* React + Vite (or Next.js)**

**\* \*UI Components:\* Tailwind CSS + Shadcn/UI**

**\* \*Backend:\* FastAPI (Python)**

**\* \*Databases:\* MySQL / PostgreSQL (SQL) + MongoDB (NoSQL)**

**\* \*State Management:\* Redux Toolkit / React Query**

**\* \*Auth:\* JWT tokens from FastAPI**

**\* \*Charts:\* Recharts / Chart.js for analytics**

## ## 2. \*Integration Goals\*

**| Area | Description |**

**| ---------------- | ---------------------------------------------------------- |**

**| Authentication | Connect login/register endpoints to JWT-based backend auth |**

**| Tender Search | Fetch and display tenders from /api/enriched-releases |**

**| AI Summarization | Display stored summaries retrieved from MongoDB |**

**| Readiness Score | Pull and visualize readiness data (0–100 %) |**

**| Workspace | Enable CRUD operations via /workspace/tenders endpoints |**

**| Analytics | Integrate spend-by-buyer charts using cached MongoDB data |**

## ## 3. \*Frontend Architecture\*

**frontend/**

**│**

**├── src/**

**│ ├── api/**

**│ │ ├── client.js # Axios setup with token interceptor**

**│ │ ├── tenders.js # Calls FastAPI endpoints**

**│ │ └── auth.js # Login/Register/Logout**

**│ │**

**│ ├── components/**

**│ │ ├── WorkspaceTable.jsx # Displays saved tenders + status**

**│ │ ├── SummaryCard.jsx # AI summary view**

**│ │ ├── ReadinessGauge.jsx # Animated score meter**

**│ │ └── SpendChart.jsx # Buyer spend visualization**

**│ │**

**│ ├── pages/**

**│ │ ├── Login.jsx**

**│ │ ├── Dashboard.jsx**

**│ │ ├── TenderSearch.jsx**

**│ │ └── Workspace.jsx**

**│ │**

**│ ├── context/**

**│ │ └── AuthContext.jsx # JWT + user state**

**│ │**

**│ └── App.jsx # Routing + layout**

**│**

**└── tailwind.config.js**

### ## 4. \*Data Flow Integration\*

**[User Interface]**

**↓**

**React (frontend)**

**↓ Axios API calls (JWT-authenticated)**

**[FastAPI Backend]**

**↓**

**SQL DB (Users, Workspace, Profiles)**

**MongoDB (Summaries, Scores, Analytics)**

## ## 5. \*Key Functional Components\*

**### 🔹 \*1. Login & Token Handling\***

**javascript**

**// auth.js**

**import axios from './client';**

**export const login = async (email, password) => {**

**const res = await axios.post('/auth/login', { email, password });**

**localStorage.setItem('token', res.data.access\_token);**

**return res.data;**

**};**

**javascript**

**// client.js**

**import axios from 'axios';**

**const instance = axios.create({ baseURL: 'http://127.0.0.1:8000' });**

**instance.interceptors.request.use(config => {**

**const token = localStorage.getItem('token');**

**if (token) config.headers.Authorization = `Bearer ${token}`;**

**return config;**

**});**

**export default instance;**

**### 🔹 \*2. Fetch & Display AI Summaries\***

**javascript**

**// tenders.js**

**export const getTenders = async () => {**

**const res = await axios.get('/api/enriched-releases');**

**return res.data;**

**};**

**jsx**

**// SummaryCard.jsx**

**export default function SummaryCard({ summary }) {**

**return (**

**<div className="p-4 bg-white rounded-xl shadow">**

**<h3 className="font-semibold mb-2">AI Summary</h3>**

**<p className="text-sm text-gray-700">{summary}</p>**

**</div>**

**);**

**}**

**### 🔹 \*3. Workspace Management\***

**javascript**

**// tenders.js**

**export const addToWorkspace = async (data) => {**

**const res = await axios.post('/workspace/tenders', data);**

**return res.data;**

**};**

**jsx**

**// WorkspaceTable.jsx**

**import { addToWorkspace } from "../api/tenders";**

**function WorkspaceTable({ tenders }) {**

**const handleAdd = async (tender) => {**

**await addToWorkspace({ tender\_id: tender.id, status: "Interested" });**

**alert("Added to workspace!");**

**};**

**return (**

**<table className="w-full text-left">**

**<thead><tr><th>Title</th><th>Deadline</th><th>Action</th></tr></thead>**

**<tbody>**

**{tenders.map(t => (**

**<tr key={t.id}>**

**<td>{t.title}</td>**

**<td>{t.deadline}</td>**

**<td>**

**<button className="btn" onClick={() => handleAdd(t)}>Add</button>**

**</td>**

**</tr>**

**))}**

**</tbody>**

**</table>**

**);**

**}**

**### 🔹 \*4. Readiness Score Visualization\***

**jsx**

**// ReadinessGauge.jsx**

**import { Gauge } from 'recharts';**

**export default function ReadinessGauge({ score }) {**

**return (**

**<div className="flex flex-col items-center">**

**<h4 className="text-gray-600 mb-1">Readiness Score</h4>**

**<Gauge**

**value={score}**

**width={180}**

**height={120}**

**startAngle={180}**

**endAngle={0}**

**domain={[0, 100]}**

**/>**

**<p className="text-sm text-gray-700 mt-1">{score}% Ready</p>**

**</div>**

**);**

**}**

**### 🔹 \*5. Analytics Dashboard\***

**jsx**

**// SpendChart.jsx**

**import { BarChart, Bar, XAxis, YAxis, Tooltip } from "recharts";**

**export default function SpendChart({ data }) {**

**return (**

**<BarChart width={400} height={250} data={data}>**

**<XAxis dataKey="buyer" />**

**<YAxis />**

**<Tooltip />**

**<Bar dataKey="total\_spend" fill="#4F46E5" />**

**</BarChart>**

**);**

**}**

## ## 6. \*Integration Testing\*

|  |  |
| --- | --- |
| **Test** | **Expected Result** |
| **Login with valid JWT** | **Redirects to dashboard** |
| **Fetch tenders** | **List renders in < 2 s** |
| **Add to workspace** | **Appears instantly in table** |
| **View summary** | **Summary text visible** |
| **Readiness score** | **Gauge renders with correct %** |
| **Analytics chart** | **Buyer data populates correctly** |

## ## 7. \*Challenges & Solutions\*

|  |  |
| --- | --- |
| **Challenge** | **Solution** |
| **CORS issues between frontend & backend** | **Configured FastAPI CORS middleware** |
| **Large AI summary data loads** | **Added pagination + lazy loading** |
| **JWT expiry** | **Auto-refresh mechanism in Axios interceptor** |
| **MongoDB aggregation delay** | **Cached analytics on backend cron job** |

**Team Members:**

* **Khethiwe Skosana - Project Lead & Backend Architecture**
* **Ashwil - Frontend Development & UI/UX**
* **Koketso Kgogo - Frontend Components & User Experience**
* **Snethemba Mthembu - Backend API & Business Logic**
* **Onthatile Kilelo - AI Integration & Data Processing**