ide 1: Introduction — Who I Am

“Hi everyone, I’m [Your Name], currently pursuing my B.Tech in Information Technology at Anna University.

During my internship with Dover’s Data & Analytics team, I worked on a project that’s not just about building dashboards — it’s about solving a real business challenge: managing and governing Power BI content across a large, distributed organization.”

“My role focused on backend development and visualization:

- I retrieved and cleaned metadata from the Power BI REST API
- Built visualizations to show usage patterns, dataset freshness, and user activity
- Implemented Row-Level Security to restrict access by OpCo
- Replicated Power BI visuals using Python to maintain consistency

The goal was to make governance insights not only accurate — but also easy to understand and act on

🛡️ **Slide 2: What Is Data Governance?**

“Before we dive into the app, let’s talk about what data governance actually means.

Data governance is the process of ensuring that data is:

- ✅ Accurate and consistent
- 🔐 Secure and accessible only to the right people
- 📌 Properly owned and maintained
- ⚙️ Optimized for performance and efficiency

In a large company like Dover, where Power BI is used across many teams and OpCos, governance is essential to avoid:

- Outdated or duplicate reports
- Unused datasets

- Uncontrolled access
- Performance issues and clutter

So governance isn't just about control — it's also about **optimization**."

🔗 **Slide 3: Why We Built This App**

"Our goal was to build a centralized, self-service web app that helps Dover:

- Monitor usage across Power BI workspaces
- Identify outdated or inactive content
- Enforce secure access using Row-Level Security
- Optimize the environment by reducing clutter and improving performance

In short: **This app helps us centralize the data governance process to keep our Power BI environment clean, secure, and optimized.**"

🧠 **Week 1: Accessed Shared Power BI Workspaces**

"I started by exploring shared Power BI workspaces using a Microsoft Learn access token. This helped me understand how workspaces are structured and what kind of metadata we could access — like reports, dashboards, datasets, and user roles."

✅ **Week 2: Connected to Power BI REST APIs**

"Next, I connected to the Power BI REST APIs to pull real metadata. I was able to retrieve reports, dashboards, datasets, users, and refresh logs — which became the foundation for all the analysis we do in the app."

✅ **Week 3: Built the Streamlit Web App**

"Once I had the data, I started building the Streamlit app. I designed a multi-page layout with clean navigation so users could easily explore different governance metrics — like reports, datasets, users, and activity trends."

✅ **Week 4: Implemented Row-Level Security (RLS)**

"To make sure access was secure, I implemented Row-Level Security. This means users only see data relevant to their OpCo — which is a key part of governance and compliance."

✅ **Week 5: Recreated Visuals Using Python**

“I then recreated key Power BI visuals using Python libraries like Plotly and Matplotlib. This allowed us to customize the visuals and make them interactive inside the Streamlit app.”

✅ **Week 6: Classified Active vs. Inactive Assets**

“Finally, I wrote logic to classify reports and datasets as active or inactive — based on refresh history and usage logs. This helps teams identify outdated or unused content and clean up the environment.”