Azure AD SSO Integration

Step-by-Step Guide

Prysmian - IT Department

Created by Dinulescu Cosmin Ovidiu (Gemini/Claude)

Prerequisites

Before starting, ensure you have:

- Azure AD Tenant with admin access
- Ability to create App Registrations
- Python environment with pip

Required package:

```
pip install msal
```

Step 1: Azure Portal Setup

1.1 Register Application:

- Go to Azure Portal (portal.azure.com)
- Navigate to Azure Active Directory
- Select App registrations
- Click New registration

Configuration:

- Name: Prysmian Induction Portal
- Supported account types: Single tenant
- Redirect URI: http://localhost:8501 (Web)

1.2 Configure Authentication:

- Go to Authentication tab
- Add platform: Web
- Add redirect URIs for production
- Enable ID tokens (Implicit grant)

1.3 Get Credentials:

- Application (client) ID - from Overview
- Directory (tenant) ID - from Overview
- Client Secret - from Certificates & secrets

Step 2: Configure Secrets

Create file: .streamlit/secrets.toml

[azure]

```
client_id = YOUR_CLIENT_ID
client_secret = YOUR_CLIENT_SECRET
tenant_id = YOUR_TENANT_ID
redirect_uri = http://localhost:8501
```

IMPORTANT:

- Never commit secrets.toml to git
- Add to .gitignore

Step 3: Enable SSO

In induction.py, add at the beginning:

```
from modules.sso_azure import require_authentication
require_authentication()
```

This will:

- Redirect to Azure login if not authenticated
- Process the OAuth callback
- Auto-save user profile from SSO token
- Allow app to continue once logged in

Step 4: Update User ID

In data_manager.py, modify get_user_id():

Add at the beginning of the function:

```
sso_user = st.session_state.get('sso_user')
```

if sso_user and sso_user.get('oid'):

return sso_user['oid']

This uses Azure Object ID for consistent tracking.

Security Notes

Production requirements:

- Use HTTPS (required for OAuth)
- Implement token refresh
- Configure proper logout URL
- Review Azure AD permissions