🚀 Complete Step-by-Step Launch Guide for SharePoint AI Dashboard

I'll guide you through launching the SharePoint AI Dashboard from scratch. Let me break this down into clear, detailed steps.

📋 Prerequisites Setup

Step 1: Install Required Software

1.1 Install Node.js

1. Go to https://nodejs.org/

2. Download the LTS version (20.x recommended)

3. Run the installer and follow the setup wizard

4. Verify installation:

node --version

npm --version

4. You should see version numbers (e.g., v20.11.0 and 10.2.4)

1.2 Install Docker

1. Go to https://www.docker.com/products/docker-desktop/

2. Download Docker Desktop for your operating system

3. Install and start Docker Desktop

4. Verify installation:

docker --version

docker-compose --version

4. You should see version numbers

1.3 Install Git (if not already installed)

1. Go to https://git-scm.com/downloads

2. Download and install Git for your system

3. Verify installation:

git --version

Step 2: Get SharePoint App Registration

2.1 Create Azure App Registration

1. Go to https://portal.azure.com/

2. Sign in with your Microsoft account

3. Navigate to Azure Active Directory → App registrations

4. Click "New registration"

5. Fill out the form:

- Name: SharePoint AI Dashboard

- Supported account types: Accounts in this organizational directory only

- Redirect URI: http://localhost:3000/auth/callback

6. Click "Register"

2.2 Configure App Permissions

1. In your new app, go to API permissions

2. Click "Add a permission" → Microsoft Graph

3. Select "Delegated permissions"

4. Add these permissions:

- Sites.Read.All

- Files.ReadWrite.All

- User.Read

5. Click "Add permissions"

6. Click "Grant admin consent" (you need admin rights)

2.3 Get App Credentials

1. Go to Overview tab and copy:

- Application (client) ID fd3b804c-5ac4-4e00-8359-f6712fc1e634

- Directory (tenant) ID a68d3c04-09fe-4a33-a02c-e880c1a7504d

2. Go to Certificates & secrets → Client secrets

3. Click "New client secret" Value: Kvr8Q~czR0iA78JhxsmBbO5xeKweLF1uKf5\_wdz.

Secret iD: 77e3b789-355f-4c98-8c81-0d3cf46462b8

4. Add description: SharePoint AI Dashboard Secret

5. Set expiration: 24 months

6. Click "Add" and COPY THE SECRET VALUE (you won't see it again!)

Step 3: Get AI Service API Key

Option A: OpenAI API Key

1. Go to https://platform.openai.com/

2. Sign up or log in

3. Go to API Keys section

4. Click "Create new secret key"

5. Copy and save the API key securely

Option B: Azure OpenAI (Alternative)

1. Go to https://portal.azure.com/

2. Create an Azure OpenAI resource

3. Get the endpoint URL and API key from the resource

🛠️ Project Setup

Step 4: Clone and Setup Project

4.1 Open Terminal/Command Prompt

- Windows: Press Win + R, type cmd, press Enter

- Mac: Press Cmd + Space, type Terminal, press Enter

- Linux: Press Ctrl + Alt + T

4.2 Navigate to Your Desired Folder

# Example: Navigate to your Documents folder

cd Documents

# Or create a new projects folder

mkdir Projects

cd Projects

4.3 Clone the Project

# Navigate to the project folder you already have

cd /Users/husseinsrour/Downloads/Sharepoint\_project

# Verify you're in the right place

ls

You should see folders like: client, server, docker-compose.yml, etc.

Step 5: Configure Environment

5.1 Create Environment File

# Copy the example environment file

cp .env.example .env.development

# Open the file for editing

# Windows:

notepad .env.development

# Mac:

open -e .env.development

# Linux:

nano .env.development

5.2 Edit Environment Configuration

Replace these values in .env.development:

# SharePoint Configuration (from Step 2)

SHAREPOINT\_CLIENT\_ID=your\_client\_id\_from\_azure

SHAREPOINT\_CLIENT\_SECRET=your\_client\_secret\_from\_azure

SHAREPOINT\_TENANT\_ID=your\_tenant\_id\_from\_azure

# AI Configuration (from Step 3)

OPENAI\_API\_KEY=sk-proj-rA-RowefD08caKiLup8jAImBklu57-dDqPUl79KEyzxmyTTKTLFteEdvF3kF\_KbpuGg9ga\_kKRT3BlbkFJ4X6sPrR79Nj0aJLnfDa6Vz9Xn6oUW5x-hNfEDYURZIC94BQGRSfouPQzQ1mtyVvNNaCkqBvaIA

OPENAI\_MODEL=gpt-3.5-turbo

# Database (keep default for development)

DB\_NAME=sharepoint\_ai\_dashboard\_dev

DB\_USER=sharepointai\_dev

DB\_PASSWORD=dev\_password\_123

# Security (generate strong secrets for production later)

JWT\_SECRET=dev\_jwt\_secret\_key\_not\_for\_production\_use

ENCRYPTION\_KEY=dev\_encryption\_key\_32\_characters

SESSION\_SECRET=dev\_session\_secret

# Other settings (keep defaults)

NODE\_ENV=development

BACKEND\_PORT=3001

FRONTEND\_PORT=3000

Save the file after making these changes.

🚀 Launch Methods (Choose One)

Method 1: Quick Docker Launch (Recommended for Beginners)

6.1 Start All Services with Docker

# Make sure you're in the project root directory

pwd

# Should show: /Users/husseinsrour/Downloads/Sharepoint\_project

# Start all services (this will take 5-10 minutes the first time)

docker-compose --env-file .env.development up --build

What this does:

- Downloads and builds all required services

- Starts database, cache, frontend, backend, and monitoring

- Shows logs from all services

6.2 Wait for Services to Start

Watch the terminal output. You'll see lots of text. Wait until you see:

sharepoint-ai-frontend | ✓ Built in X seconds

sharepoint-ai-backend | Server running on port 3001

sharepoint-ai-db | database system is ready to accept connections

6.3 Run Database Setup (New Terminal)

Open a new terminal/command prompt (keep the first one running):

# Navigate to project directory

cd /Users/husseinsrour/Downloads/Sharepoint\_project

# Run database migrations

docker-compose --env-file .env.development exec backend npm run migrate

You should see:

✓ Connected to database

✓ Migrations table ready

✓ All migrations completed successfully!

Method 2: Development Setup (For Developers)

6.1 Install Dependencies

# Install backend dependencies

cd server

npm install

# Install frontend dependencies

cd ../client

npm install

# Go back to project root

cd ..

6.2 Start Development Databases

# Start only database and cache services

docker-compose -f docker-compose.test.yml up -d postgres redis

6.3 Setup Database

cd server

npm run migrate

6.4 Start Development Servers

Terminal 1 (Backend):

cd server

npm run dev

Terminal 2 (Frontend):

cd client

npm start

✅ Verify Everything is Working

Step 7: Check the Application

7.1 Access the Application

Open your web browser and go to:

- Frontend: http://localhost:3000 (development) or http://localhost (Docker)

- Backend API: http://localhost:3001/health

7.2 Test Health Endpoints

Open these URLs in your browser:

Backend Health Check:

http://localhost:3001/health

Should show: {"status":"ok","timestamp":"..."}

Frontend Health Check:

http://localhost/health

Should show a simple health status page.

7.3 Check All Services (Docker Method Only)

# Check all services are running

docker-compose --env-file .env.development ps

Should show all services as "Up" or "healthy".

Step 8: Access Monitoring Tools (Docker Method Only)

If you used Docker, you also have monitoring tools available:

- Grafana (Monitoring Dashboards): http://localhost:3002

- Username: admin

- Password: dev\_admin

- Kibana (Logs): http://localhost:5601

- No login required for development

- Prometheus (Metrics): http://localhost:9090

- No login required

🎯 Next Steps - Using the Application

Step 9: First Login

9.1 Access the Frontend

1. Go to http://localhost:3000 (or http://localhost if using Docker)

2. Click "Sign in with Microsoft"

3. You'll be redirected to Microsoft login

4. Sign in with your SharePoint/Office 365 account

5. Grant permissions when prompted

6. You'll be redirected back to the dashboard

9.2 Explore Features

Once logged in, you can:

- View document analytics

- Upload and analyze documents with AI

- Configure settings

- View user activity logs

- Export reports

🔧 Troubleshooting Common Issues

Issue 1: "Port already in use"

Problem: Error like Port 3001 is already in use

Solution:

# Find what's using the port

lsof -i :3001

# Kill the process (replace PID with the actual process ID)

kill -9 PID

# Or change the port in .env.development

BACKEND\_PORT=3002

Issue 2: "Cannot connect to database"

Problem: Database connection errors

Solution:

# Check if Docker is running

docker ps

# Restart database service

docker-compose --env-file .env.development restart database

# Check database logs

docker-compose --env-file .env.development logs database

Issue 3: "SharePoint authentication failed"

Problem: Can't sign in to SharePoint

Solution:

1. Double-check your SHAREPOINT\_CLIENT\_ID, SHAREPOINT\_CLIENT\_SECRET, and SHAREPOINT\_TENANT\_ID

2. Verify redirect URI in Azure: http://localhost:3000/auth/callback

3. Make sure you granted admin consent for API permissions

Issue 4: "AI service not responding"

Problem: AI analysis doesn't work

Solution:

1. Check your OPENAI\_API\_KEY is correct

2. Verify you have OpenAI API credits

3. Try changing OPENAI\_MODEL to gpt-3.5-turbo if using gpt-4

🛑 How to Stop the Application

Docker Method:

# Stop all services

docker-compose --env-file .env.development down

# Or stop but keep data

docker-compose --env-file .env.development stop

Development Method:

- Press Ctrl + C in each terminal running the servers

📊 Application URLs Summary

Once everything is running:

| Service | URL | Purpose |

|--------------------|--------------------------------|---------------------------------|

| Main Application | http://localhost:3000 | Use the SharePoint AI Dashboard |

| API Documentation | http://localhost:3001/api-docs | View API endpoints |

| Backend Health | http://localhost:3001/health | Check backend status |

| Grafana Monitoring | http://localhost:3002 | View performance metrics |

| Kibana Logs | http://localhost:5601 | View application logs |

| Prometheus | http://localhost:9090 | Raw metrics data |

🎉 You're All Set!

You now have a fully functional SharePoint AI Dashboard running locally! The application provides:

- 📊 Analytics Dashboard - View document usage statistics

- 🤖 AI Document Analysis - Analyze documents with OpenAI/GPT

- 🔗 SharePoint Integration - Connect to your SharePoint sites

- ⚙️ Settings Management - Configure AI models, UI, and preferences

- 📈 Real-time Monitoring - Track performance and usage

- 🔒 Enterprise Security - JWT authentication and rate limiting

Start by signing in with your Microsoft account and exploring the dashboard!