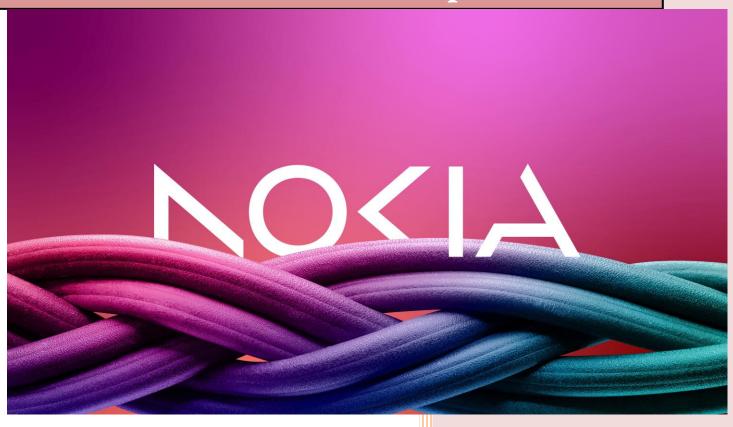# 2025

Nokia KPI Report Task

Done by : Hania Ahmed Helmy

Presented to : Eng.Mohamed Nabil

9/14/2025

Nokia KPI Report App

Overview

This application is a full-stack system built to generate Excel reports, export them to PowerPoint, and send them automatically via email scheduling.

It integrates FastAPI (backend) with React + Material-UI (frontend), and supports user authentication, scheduling, and automated email reporting.

The system allows:

- Uploading and cleaning raw datasets.

- Generating aggregated KPI Excel reports.

- Exporting charts to PowerPoint.

- Scheduling email delivery of reports.

- Managing settings via a modern web interface.

How to Run the Application

- Backend (FastAPI): Run ' uvicorn app:app ' from the backend folder.
  - Access it at http://localhost:8000/docs
- Frontend (React): Run 'npm install' (first time only), then 'npm run dev' from the frontend (settings-ui) folder.
  - Access it at http://localhost:5173

Environment Configuration (Please add the following .env files)

Backend .env:

DATASET_PATH=./dataset.csv

OUTPUT_DIR=./out

Gmail SMTP settings

SMTP_HOST=smtp.gmail.com

SMTP_PORT=465

SMTP_USER=haniahelmy20@gmail.com

SMTP_PASS=xzpkklfcdssdxdqk

SMTP_FROM=haniahelmy20@gmail.com

Database + App

DATABASE_URL=postgresql://neondb_owner:npg_Kl2hV6dDzCou@ep-orange-shape-adp1vijl-pooler.c-2.us-east-1.aws.neon.tech/neondb?sslmode=require

SECRET_KEY=a4f8e2d9c76b4f7b83df45f9ad8e33a2e2c0d423f45e1c99f33bbfc01a9f11cd

CORS_ORIGIN=http://localhost:5173

TOKEN_MINUTES=1440

Frontend .env:

VITE_API_URL=http://localhost:8000

Installation Requirements

- **Backend**: pip install -r requirements.txt
- **Frontend**: npm install

Data Cleaning & Aggregation

The raw dataset is cleaned and transformed before being used for reporting.

1. **Standardizing Columns**

   o The first two columns are renamed to date and sitecode.

   o Dates are parsed with day-first format (e.g., 31/12/2024).

2. **Handling Missing Data**

   o If all KPIs are missing for a row, that row is dropped.

   o Missing KPI values are filled as follows:

      ▪ If an entire week is missing for a KPI → fill with the global minimum for that KPI.

      ▪ If only some values are missing in a week → fill missing values with the minimum of that week.

3. **Aggregation**

   o After filling missing values, data is grouped per date.

   o All KPIs across sites for the same date are summed together.

   o The result is a clean, aggregated dataset ready for analysis.

4. **Output**

   o The cleaned dataset is saved as backend/cleaned_dataset.csv.

   o This file is the source for generating Excel reports and PowerPoint charts.

Application Flow

1. **Authentication**

   • Users register and log in securely.

   • Passwords are encrypted using bcrypt.

   • JWT tokens stored in HTTP-only cookies for protection.

2. **Settings Management**

   • Settings are persisted in settings.json.

   • When a user logs back in, their previous settings are remembered.

   • Swagger (/docs) can also be used to insert initial values into the database.

3. **Report Generation**

   • Excel is regenerated automatically whenever settings change.

- PowerPoint must be generated manually with the Save PPT button, so users can open the excel file that the code generated and style it/format it in any way they like (Gave user the option instead of forcing a format they might be unhappy with (Found that that was the more user friendly approach))

4. Email Sending

- SMTP configured with Gmail (haniahelmy20@gmail.com).

- Reports are attached as PowerPoint presentations.

5. Scheduling

- APScheduler runs scheduled jobs.

- Emails can be sent daily, weekly, or monthly.

- Scheduler runs only while the backend server is active.

How to Use the App
- ✓ Login with: Email-> haniahelmy20@gmail.com , Password -> Hania12345
- ✓ Configure settings step by step.
- ✓ Export PowerPoint with Save PPT icon.
- ✓ Send Now or Schedule in the final step.

Security
- o Passwords are hashed with bcrypt.
- o Tokens stored securely.
- o SMTP credentials in .env.

Future Work & Deployment
1. Add per-user settings table in database.
2. Deploy on a server to keep scheduler running even if local machine is off.