Individual – SmartDataProject User-guide

Esteban BARRACHO

15 juillet 2025

POLY-TECH
ENGINEERING

Table des matières

1 Preamble 3

1.1 Introduction 4

1.2 What is this software ? 4

1.2.1 Its objective 4

1.2.2 Key features 4

1.2.3 What it is not ! 5

1.2.4 For which audience ? 5

1.2.5 Licence of use 5

1.3 Legal notices 6

1.4 How to install it ? 7

1.4.1 Where is the installation package ? 8

1.4.2 Requirements 8

1.4.3 Supported Operating Systems 8

1.4.4 Server Mode 8

1.4.5 Minimum System Requirements 8

1.4.6 Installation Path 8

1.4.7 Environment Variables 8

1.4.8 Modifications 8

1.4.9 Additional Tools (Optional) 8

1.4.10 Launch from Docker (Recap) 9

1.5 How to use it ? 10

1.5.1 By features 10

1.5.2 Start here 10

1.5.3 By use case 11

1.5.4 What to do in the event of errors ? 11

1.6 Frequently Asked Questions (FAQ) 12

1.7 Installation and Configuration 12

1.8 General Use 12

2 Glossaire 13

3 Bibliographie 14

3.1 Further reading 14

3.2 Structures to compensate for pre-requisites 14

4 Index 14

5 Appendix 15

5.1 Appendix A : Future Plans 15

1 Preamble

This document is the official user guide for the **Smart Data Project**, a professional-grade platform developed independently as part of a real-world collaboration with an external company during the Master's programme in Software Engineering.

The Smart Data Project offers an integrated digital workspace designed to optimize project management, collaborative planning, and administrative traceability. It combines a secure backend powered by FastAPI with a modular web interface tailored to technical teams, consultants, and administrative managers. Key features include :

- a dynamic agenda synchronized with Microsoft Outlook via the Azure API,
- an intelligent task encoder powered by AI (OpenRouter + Mistral),
- full CRUD management of projects, invoices, offers, phases, and time tracking,
- Excel import functionality with automatic structure suggestions by AI,
- mobile-ready user interface with real-time alerting on time logs.

This user guide provides detailed instructions for using each module and explains the underlying logic of the system. Beyond being a manual for end-users, it also highlights the architectural choices, third-party integrations, and technical challenges encountered during development.

The system aims to be robust, maintainable, and extensible, providing a strong foundation for future integration with external systems, intelligent document parsing, and automated planning.

1.1 Introduction

This document serves as the user guide for the **Smart Data Project**, a professional web application developed as part of an industrial collaboration during the Master in Software Engineering. It is designed to support collaborative project management, employee planning, time tracking, and administrative reporting in a unified and intelligent environment.

The Smart Data Project integrates a modular backend developed in Python with FastAPI, a responsive frontend, and several smart features such as Outlook synchronization via the Azure API, intelligent Excel import using OpenRouter and Mistral, and AI-powered structure correction for database entries.

This guide provides a comprehensive overview of the platform's core functionalities. Users will learn how to authenticate, create and manage projects, assign tasks, log work hours, track invoices, and automate planning through a combination of manual entries and AI suggestions.

Aimed at administrative managers, consultants, developers, and end-users from any technical background, the guide walks through each module and interface screen with practical examples and clear instructions. It also emphasizes how the platform fosters traceability, real-time collaboration, and seamless integration with tools such as Microsoft Outlook.

By following this guide, users will gain a solid understanding of how to use Smart Data Project efficiently in real business environments and how to extend or adapt it for future needs.

1.2 What is this software ?

This software is a smart collaborative management platform designed for operational teams and administrative staff. It combines planning, time tracking, project supervision and document traceability into a single web-based interface. Powered by FastAPI and intelligent Excel interpretation using OpenRouter and AI models, it streamlines project-related operations from planning to billing.

1.2.1 Its objective

The **Smart Data Project** aims to facilitate structured and efficient management of projects, personnel, and data flows within an enterprise context. It introduces intelligent automation to reduce manual encoding efforts and enhance accuracy through Outlook synchronization and AI-suggested data adaptation.

1.2.2 Key features

- Synchronisation with Microsoft Outlook via Azure API to fetch and display collaborative agendas.
- AI-assisted import of Excel files : intelligent correction and alignment of tabular data to match SQL schemas.
- Full CRUD support for projects, tasks, clients, phases, offers, invoices, and prestations.
- Real-time administrative dashboard and traceability interface.
- Role-based access control and time-sensitive alerts for encoding delays.

1.2.3 What it is not !

While **Smart Data Project** is rich in features and smart integrations, it also has clearly defined limits :

Not a full ERP system It does not include accounting, payroll, or inventory management functionalities typically found in enterprise resource planning software.

Not a big data analytics platform Though capable of handling typical business loads, the platform is not intended for large-scale data mining, nor does it integrate Hadoop-like ecosystems.

Not a chatbot or generative AI assistant The AI module is restricted to suggesting data alignment corrections and extracting task information from structured content like emails or spreadsheets.

Not a file hosting system Files can be linked or referenced, but the software does not function as a document management or storage solution (like Google Drive or SharePoint).

Not a substitute for human validation AI suggestions and automatic mappings require validation. Human review is essential before any critical insertion or deletion.

1.2.4 For which audience ?

Knowledge, skills, roles The software is designed for :

- Project managers and coordinators in engineering or consulting companies.
- Administrative users managing client interactions, time logs, and billing.
- Developers looking to extend or integrate the platform in new workflows.

Basic knowledge of project coordination, data entry and collaborative tools (like Outlook or Excel) is assumed. Interfaces are designed to remain accessible for non-technical users.

1.2.5 Licence of use

The **Smart Data Project** is distributed under a **"Private Code License"** with restricted usage rights defined by the project author.

Attribution required Any reuse or redistribution must credit the author, **Esteban Barracho**, and include the original license terms as stated in the distributed repository.

Modification restricted Modifications are allowed for private use only. Public redistribution of modified versions is prohibited unless explicitly authorized by the author.

Commercial use limited The code and platform are not to be sold or incorporated into commercial platforms without a negotiated license agreement.

No warranty The software is provided "as-is", without warranty of any kind. The author is not liable for any damages or losses caused by the use or misuse of the platform.

This licensing approach ensures ethical use, protects intellectual property, and maintains the quality and coherence of the distributed version.

1.3 Legal notices

Privacy and data processing The Smart Data Project is committed to respecting the privacy and confidentiality of its users. In accordance with the General Data Protection Regulation (GDPR), all user data is treated securely. Sensitive information such as session identifiers or tokens is stored locally on the user's device (via cookies), and no personal data is sold or shared with third parties.

Data retention and usage The data handled by the platform (e.g., project details, time logs, tasks, Outlook calendars) is stored in a local database accessible only to authenticated users and administrators of the system. The data is retained only for operational and administrative purposes and remains under the control of the data processor (e.g., the enterprise deploying the system).

Personal information and authentication Using Smart Data Project may involve identification (login via email and 2FA code) and the creation of a user profile. However, this information is only used for session management and audit trail functionalities. The system does not request unnecessary personal data and does not implement profiling or behavioural tracking.

Integration with Microsoft Outlook The Outlook calendar integration via Azure OAuth2 is strictly limited to event synchronization for planning purposes. No content of emails or calendar metadata is collected beyond what is necessary to match tasks or populate the user's calendar view. The token remains stored locally (as a secure cookie) and is revocable at any time by the user.

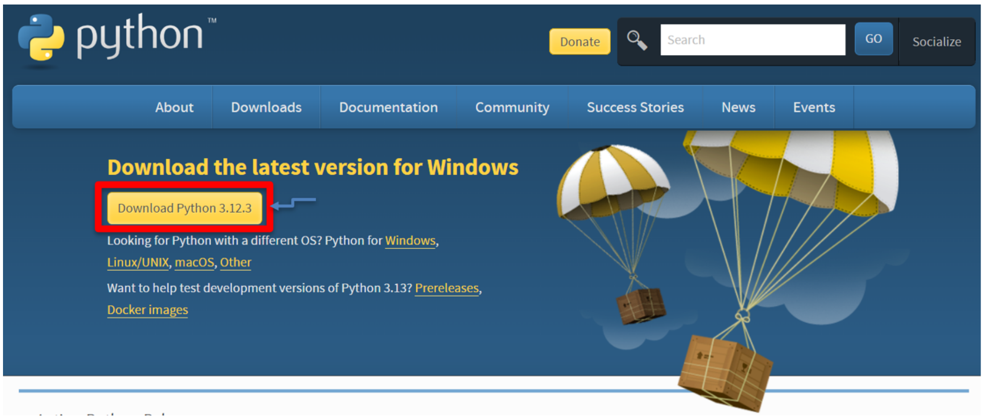
Disclaimers The author of the Smart Data Project, Esteban Barracho, cannot be held responsible for any misuse, data loss, or erroneous results obtained from the platform. Users are responsible for verifying the correctness of their entries and the relevance of any automated suggestions or AI adaptations.

GDPR compliance The platform implements access control, token expiration, and encrypted session cookies. Combined with its local deployment model, this ensures full compliance with GDPR regarding data minimization, purpose limitation, and user control. No data is transmitted to external servers beyond explicitly authorized API endpoints (e.g., OpenRouter or Microsoft Graph when configured).

1.4 How to install it ?

To install and run the Smart Data Project locally, follow these steps carefully :

1. Install Python Start by installing Python 3.10+ from the official website : Python. Make sure to check `Add Python to PATH` during installation as shown in the photo :



Download the latest version for Windows

Looking for Python with a different OS? Python for Windows, Linux/UNIX, macOS, Other

Want to help test development versions of Python 3.13? Prereleases, Docker images

FIGURE 1 – Python Installer with PATH option

2. Install Docker (Optional but Recommended) If you want to deploy the application with all services (backend + database) easily :

- Download and install Docker Desktop from : Docker
- Enable virtualization in your BIOS if needed.
- Make sure Docker is running before proceeding.

3. Open the project folder Navigate to the directory `PolyBase/code/polybase`. To do this on Windows :

- Go to the folder in Explorer.
- Right-click on `polybase` and choose `Open in Terminal`.

4. Install dependencies Run the following command to install all required packages :

```
pip install -r requirements.txt
```

5. Launch the application In Docker Run the application with the command :

```
docker-compose up --build
```

After a few seconds, the application will be accessible in your browser via : `http://127.0.0.1:8000/` or `http://localhost:8000/` depending on configuration.

1.4.1 Where is the installation package ?

The list of required packages is defined in `requirements.txt` located in `code/polybase/`.

1.4.2 Requirements

Operating systems Smart Data Project is developed for Windows 10+, but compatible with Linux/macOS via Python and Docker.

Dependencies Handled automatically using pip. See `requirements.txt` for full list :

- FastAPI, Uvicorn
- SQLAlchemy, pymysql
- Pandas, Openpyxl
- Requests, MSAL, Levenshtein

1.4.3 Supported Operating Systems

- Windows 10 / 11 (recommended)
- Ubuntu 22.04 LTS
- macOS Ventura or later

1.4.4 Server Mode

The app runs in local mode, no external server is required. All interactions, including with Outlook and OpenRouter, are done securely.

1.4.5 Minimum System Requirements

- **CPU :** Intel Core i3 or better
- **RAM :** 4 GB minimum (8 GB recommended)
- **Disk :** 500 MB for code and data
- **Graphics :** Onboard graphics sufficient

1.4.6 Installation Path

The application can be placed anywhere on your computer. Just remember its path to run it.

1.4.7 Environment Variables

None required manually. The application uses a `.env` file stored under `polybase/.env`.

1.4.8 Modifications

If you want to test with another dataset or change login behavior, refer to :

```
code/polybase/app/main.py
```

1.4.9 Additional Tools (Optional)

For database inspection :

- Use a MySQL GUI client (e.g. DBeaver or MySQL Workbench)
- Credentials can be found in `.env`

1.4.10 Launch from Docker (Recap)

In `code/polybase`, run :

```
docker-compose down -v & docker-compose up --build
```

1.5 How to use it ?

1.5.1 By features

Systematic overview of main features The Smart Data Project offers a secure and user-friendly web interface to manage professional tasks, hours, invoices, and schedules. Below are its key features :

Login and Dashboard Users authenticate via a login form. After entering credentials and a second-factor code, they are redirected to a secure dashboard with personalized access.

Manual or Task-linked Time Entry Navigate to the `/prestation` or `/agenda` page to record hours :

- Manual entry : specify date, project, time spent, and description.
- Task entry : select a task from the list and add time accordingly.

Alerts are automatically triggered for delayed entries (`>2 days` or `>7 days`).

Project and Client Management Admin users can access `/projects` and `/clients` to add, edit, or delete projects and clients. Each entry is validated and linked to related invoices or phases.

Smart Import (Excel) From the admin interface, users can upload Excel files :

- Structure is adapted automatically to the target SQL table.
- The AI (OpenRouter/Mistral) suggests improvements or flags issues.
- After review, valid entries are inserted into the database.

Outlook Agenda Sync When connected, users can synchronize their Outlook calendar with the local agenda. Tasks detected in emails or events can be converted into tracked entries via OpenAI-assisted NLP.

Export to Excel Any table (tasks, prestations, invoices, etc.) can be exported by clicking the export button in the admin interface. It generates an `.xlsx` file matching the table schema.

1.5.2 Start here

A quick guide for beginners

1. Launch the application via `docker-compose up`.
2. Visit `http://localhost:8000` in your browser.
3. Log in with your credentials (use demo credentials if available).
4. Explore pages : dashboard, agenda, tasks, prestations.
5. Try inserting an entry (manual or via Outlook).
6. Import a sample Excel from `/admin/import`.
7. Export a table to validate structure.

1.5.3 By use case

Example : Encoding Weekly Timesheets

- Go to `/agenda`.
- Click "New Time Entry".
- Select project "Internal Dev", enter 3h on Monday with description.
- Entry appears on agenda, and exportable via admin panel.

Example : Importing external Excel invoices

- Navigate to `/admin/import`.
- Choose table "Facture" and upload your Excel file.
- Review AI suggestion for columns.
- Submit and verify entries.

1.5.4 What to do in the event of errors ?

Error management The application logs every major action and failure in the server console.

- For login failures : ensure credentials are correct.
- For Excel imports : review AI feedback before applying structure.
- For API errors (422 or 500) : check the logs or verify the submitted data.

Most validations are client-side and supplemented with custom HTML error templates.

1.6 Frequently Asked Questions (FAQ)

1.7 Installation and Configuration

- **Q : How do I install the Smart Data Project ?**
- A : Follow the installation instructions provided in the documentation. You can either run `main.py` directly with Python or use Docker via `docker-compose up -build`.
- **Q : What are the system requirements ?**
- A : The application is tested under Windows 10+, with a minimum of 4GB RAM and 500MB of free disk space. Python 3.8+ or Docker Desktop is required.
- **Q : How do I install Docker and run the project ?**
- A : Download Docker Desktop from docker.com, then right-click on the folder `PolyBase\code\polybas` and choose "Open in Terminal". Then execute `docker-compose up -build`.
- **Q : Where is the list of dependencies ?**
- A : The list of Python packages is located in `requirements.txt`.
- **Q : Can I run it without Docker ?**
- A : Yes. As long as Python 3.8+ is installed, simply run :
  ```
  pip install -r requirements.txt
  ```
  then launch with :
  ```
  uvicorn main:app -reload
  ```

1.8 General Use

- **Q : How do I record worked hours ?**
- A : Go to the `/prestation` page, then fill in the form (date, project, hours, description). You can also encode directly from the agenda.
- **Q : How do I import Excel files ?**
- A : Admins can access the import section via `/admin/import`, choose the target table (e.g., `Facture`), upload an Excel file, and validate AI suggestions before insertion.
- **Q : Can I view Outlook events in the application ?**
- A : Yes. After connecting your Outlook account (OAuth2), events are synchronized with `/agenda`. You can link them to tasks or convert them into prestations.
- **Q : What happens if I upload a file with wrong column names ?**
- A : The OpenRouter AI will analyze the file and suggest structure corrections. You'll be able to preview and adjust the structure before inserting data into the database.
- **Q : Where can I export my data ?**
- A : All main tables (projects, clients, prestations, etc.) can be exported from the admin interface via the export buttons (Excel format).

2 Glossaire

FastAPI Framework web moderne, rapide et asynchrone pour créer des API avec Python. Il est utilisé dans Smart Data pour gérer les routes backend, l'authentification et les requêtes vers la base de données.

OpenRouter Passerelle API utilisée pour interagir avec des modèles d'intelligence artificielle (comme Mistral) de manière centralisée. Smart Data utilise OpenRouter pour suggérer des corrections de structure lors de l'import de fichiers Excel.

Session SQLAlchemy Instance de session créée par SQLAlchemy pour interagir avec la base de données. Chaque appel à la base de données dans Smart Data se fait à travers une session injectée via FastAPI.

OAuth2 Protocole d'authentification sécurisé permettant aux utilisateurs de se connecter via des services tiers (comme Outlook). Dans Smart Data, OAuth2 est utilisé pour connecter et synchroniser le calendrier Outlook personnel de l'utilisateur.

Agenda Outlook Calendrier personnel synchronisé via Microsoft Graph API. Dans Smart Data, les événements Outlook sont automatiquement intégrés dans la vue `/agenda` après authentification.

Prestation Action d'encodage d'un travail accompli, incluant la date, le projet associé, la durée et une description. Dans Smart Data, ces prestations peuvent être ajoutées manuellement ou dérivées de tâches/événements Outlook.

Levenshtein Algorithme de distance d'édition utilisé pour mesurer la similarité entre deux chaînes de caractères. Smart Data l'utilise pour effectuer des recherches approximatives dans toutes les tables en cas d'erreur de frappe de l'utilisateur.

Docker Plateforme permettant de contenir et d'exécuter des applications dans des environnements isolés appelés conteneurs. Dans Smart Data, Docker simplifie le déploiement en encapsulant l'environnement logiciel complet.

Docker Compose Outil permettant de lancer plusieurs conteneurs Docker via un fichier YAML. Smart Data l'utilise pour orchestrer les services (API, base de données) dans le fichier `docker-compose.`

Database schema Structure logique représentant l'organisation des données dans la base relationnelle (tables, relations, types). Smart Data ajuste automatiquement les données Excel importées à ce schéma via l'IA.

3 Bibliographie

3.1 Further reading

Références

[1] François Chollet. *Deep Learning with Python*. Second Edition, Manning Publications, 2021.

[2] Casey Wilson, et al. *Applied Artificial Intelligence : A Handbook for Business Leaders*. O'Reilly Media, 2023.

[3] Andreas C. Müller and Sarah Guido. *Introduction to Machine Learning with Python : A Guide for Data Scientists*. O'Reilly Media, 2021.

[4] Carlo Batini and Monica Scannapieco. *Data Quality : Concepts, Methodologies and Techniques*. Springer, 2006.

3.2 Structures to compensate for pre-requisites

Références

[1] Luciano Ramalho. *Fluent Python : Clear, Concise, and Effective Programming*. O'Reilly Media, 2015.

[2] David Beazley. *Python Essential Reference*. Fourth Edition, Addison-Wesley, 2009.

[3] Laurent Roussel. *FastAPI : Développez des API performantes avec Python*. ENI Éditions, 2021.

[4] Peter Vass. *Microsoft 365 and Azure : Integrating Cloud Services*. Packt Publishing, 2020.

4 Index

Getting Started Guide, 4

Introduction, 3

Privacy, 6

Software features, 3

5 Appendix

5.1 Appendix A : Future Plans

In the Smart Data project, several enhancements are envisioned to further improve user experience and analytical capabilities. Notably, the following features are planned for future development :

- **Outlook API integration and task prediction** : Through the existing `outlook_sync.py` module and Microsoft Azure authentication, the system will be extended to parse user calendar events and email subjects. The goal is to identify actionable items and intelligently convert them into tasks stored in the Smart Data database.
- **AI assistance on incoming Outlook data** : Using OpenRouter-connected models such as Mistral, the application will analyze the semantic content of emails to extract relevant information (e.g., deadlines, descriptions, responsible persons) and suggest appropriate data insertion in the project's relational schema.
- **Automatic relation to projects or phases** : If an imported task via Outlook matches existing entities (based on Levenshtein similarity or named entity recognition), it will be automatically linked to the appropriate project or phase in the database, reducing the need for manual intervention.

These improvements aim to create a seamless bridge between communication tools (like Outlook) and structured project management, all supported by real-time AI-based data correction and suggestion features.