D9.5 – Updated BIG-MAP App Store

VERSION

VERSION	DATE
2.0	05-12-2023

PROJECT INFORMATION

GRANT AGREEMENT NUMBER	957189
PROJECT FULL TITLE	Battery Interface Genome - Materials Acceleration Platform
PROJECT ACRONYM	BIG-MAP
START DATE OF THE PROJECT	1/9-2020
DURATION	3 years
CALL IDENTIFIER	H2020-LC-BAT-2020-3
PROJECT WEBSITE	big-map.eu

DELIVERABLE INFORMATION

WP NO.	9
WP LEADER	Marzari (EPFL)
CONTRIBUTING PARTNERS	EPFL, KIT, DTU
NATURE	Open Research Data Pilot
AUTHORS	Tushar Thakur, Francois Liot, Giovanni Pizzi, Nicola Marzari, Tejs Vegge
CONTRIBUTORS	Same as authors
CONTRACTUAL DEADLINE	31-12-2023
DELIVERY DATE TO EC	13-12-2023
DISSEMINATION LEVEL (PU/CO)	PU

ACKNOWLEDGMENT

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 957189. The project is part of BATTERY 2030+, the large-scale European research initiative for inventing the sustainable batteries of the future.

ABSTRACT

BIG-MAP has developed a platform to share and promote its state-of-the-art tools and methods: the BIG-MAP App Store (https://big-map.github.io/big-map-registry/). This online portal serves as the primary registry of all the apps used and developed in the projects funded by BIG-MAP, offering a one stop solution to explore powerful apps for battery research. One of the aims of the App Store is to increase the accessibility and exposure of the apps, which cover various aspects of battery design, testing, simulation, and optimization. The majority of the apps are open-source and their source code is publicly available, allowing anyone to use and/or modify them for their own or collaborative research or educational purposes. The App Store provides links to not only each app's homepage, documentation and source code, but also to video tutorials to help the installation process and demonstrate typical use cases of each app. Currently, the App Store hosts 27 open-source and 2 apps developed by industrial partners.

CONTENTS

1. Introduction

Materials design and discovery has emerged as one of the most important fields of study in the past decades. This is especially true in the landscape of battery research and towards this end, several computational tools have been developed by BIG-MAP partner institutes. However, the efficient dissemination of such tools and methodologies to the broader scientific community and general public is a challenge that often gets ignored.

The BIG-MAP App Store, accessible at https://big-map.github.io/big-map-registry/, is a response to this challenge. It serves as a central repository designed to streamline access to and utilization of these computational tools. Currently it hosts 29 apps from not only BIG-MAP academic partners like EPFL, DTU, KIT, UU etc, but also from industrial partners like Solvay and Atinary Solutions creating a rich platform that includes contributions from both sectors. These apps cover a wide range of battery research ranging from purely theoretical topics such as running electronic structure calculation by automating workflows with popular codes like Quantum ESPRESSO and VASP, to experimental data analysis and visualization.

2. BIG-MAP App Store

2.1 Overview

As explained in the report D9.1, the BIG-MAP App Store includes the three following items:

1. A public website (https://big-map.github.io/big-map-registry/) that serves as a registry of BIG-MAP applications; it lists available apps and their description.
2. A metadata schema (https://github.com/BIG-MAP/big-map-registry/blob/main/README.md) that describes various aspects of BIG-MAP apps, including general scope, documentation, source code repository, etc.
3. A GitHub repository (https://github.com/BIG-MAP/big-map-registry) in open access that contains the website's source code; this is where app registration takes place via a simple pull-request.

Note that the App Store website can be reached from the BIG-MAP website (https://www.big-map.eu/), as illustrated in Figure 1.

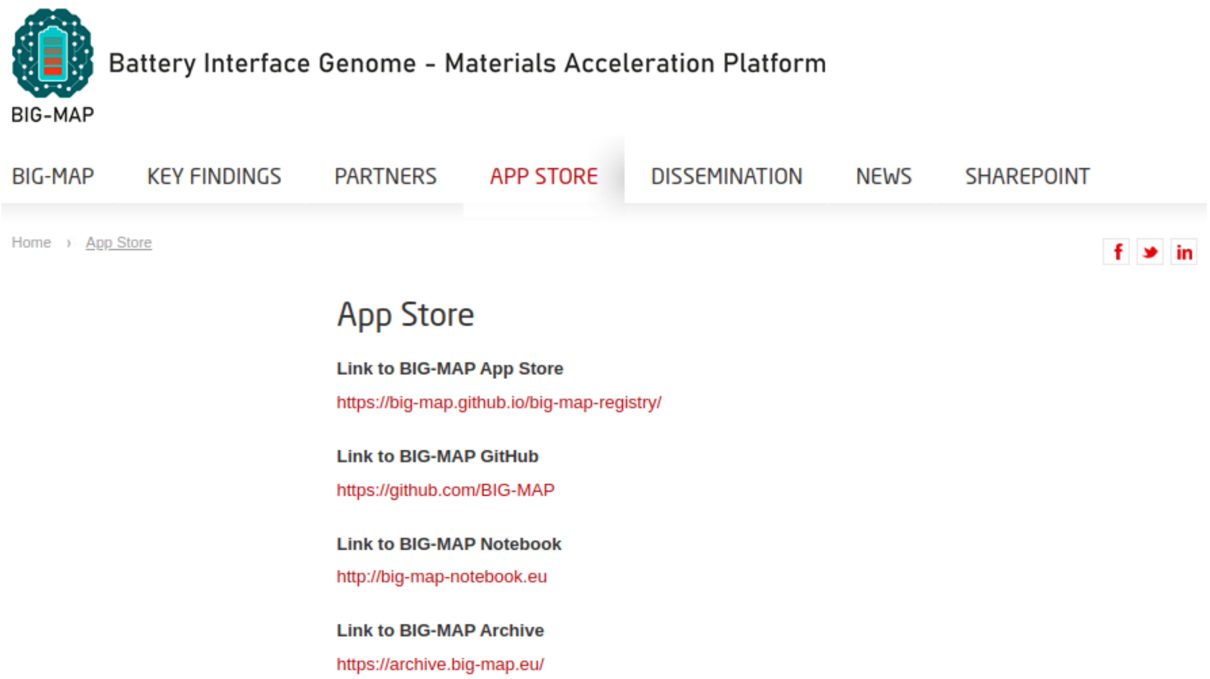

Figure 1. Homepage of BIG-MAP showing links to the App Store and the GitHub repository which serves as its source code.

2.2 Major technical overhaul

To begin with, we would like to thank the BIG-MAP community for their valuable feedback on the App Store. Their suggestions were carefully discussed and were instrumental in improving the App Store.

The latest version of the App Store introduces a number of major technical overhauls with respect to the prototype presented in the previous report. We discuss these points in detail in the following subsections.

Video tutorials

An effective way to convey usability of each registered app is to have two videos on the App Store, explaining how to install and how to use the app. This makes sure that the prospective users can get a good feeling of the capabilities at play. App authors/developers/PIs provided us with their videos (maximum length of 4 minutes, each) showing:

1. The app running under working conditions and being used, starting with a short explanation of the goal of the app and what it is trying to solve.
2. How the app can be downloaded, installed, and put in working conditions.

After an app owner has uploaded their two video files (usually in the MP4 format) to a designated Google Drive folder, the App Store source code is modified to enable anyone to play these videos directly on the website. Additional video controls, like pause and volume, are available out of the box.

As shown in Figure 2, the installation and demonstration videos of CleaseGUI are easily accessible from the app's main page.

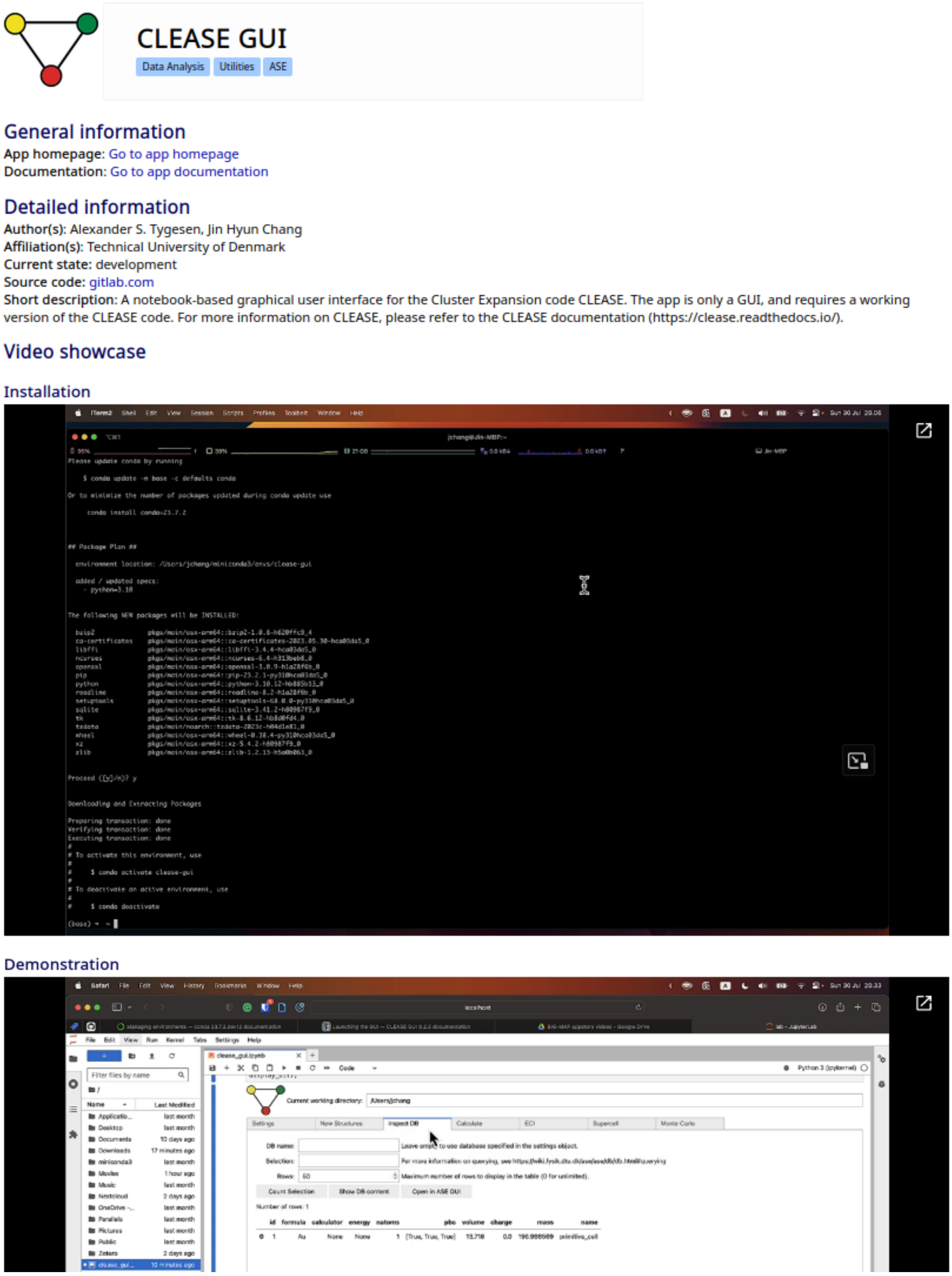

Figure 2. The main page of CleaseGUI showing app details including the tutorial videos.

Updated metadata schema

As explained in the previous report D9.1, an app is added to the registry by inserting an extra entry to `apps.yaml`. For instance, consider the CleaseGUI entry:

```yaml
clease-gui:
    git_url: https://gitlab.com/computationalmaterials/clease-gui
    metadata:
        title: CLEASE GUI
        description: |
            A notebook-based graphical user interface for the Cluster Expansion code CLEASE.
            The app is only a GUI, and requires a working version of the CLEASE code.
            For more information on CLEASE, please refer to the CLEASE documentation (https://clease.readthedocs.io/).
        authors: Alexander S. Tygesen, Jin Hyun Chang
        affiliations: Technical University of Denmark
        external_url: https://gitlab.com/computationalmaterials/clease-gui
        documentation_url: https://clease-gui.readthedocs.io
        logo: https://gitlab.com/computationalmaterials/clease-gui/-/raw/master/clease_gui/assets/clease_logo.png
        state: development
        video_installation_url: https://drive.google.com/file/d/1OSRTR3761hs1TGyk5_oMPfNyErnc6whn/preview
        video_demonstration_url: https://drive.google.com/file/d/1AITCUjLnbsT2W6WIBgCqFCMuLUXALSNy/preview
    categories:
        - data-analysis
        - utilities
        - technology-ase
```

Figure 3. Illustration of the `yaml` schema with the keywords used in the CleaseGUI app.

Descriptions of apps in the App Store are based on a metadata schema that describes various aspects of the apps, including their general scope, their source code repository, their authors, etc. An exhaustive list of available keys is shown below:

Valid keys for app entries in `apps.yaml`

Key	Requirement	Description
`metadata`	**Mandatory**	General description of the app (see below).
`categories`	**Mandatory**	An array of categories, where each category must be one of the categories specified in `categories.yaml`.
`git_url`	**Mandatory**	Link to the source code, can be github or gitlab or any other publicly available repository.

Figure 4. Illustration of the table as seen on the README of the App Store repository with the valid keys for `apps.yaml` file.

Valid keys for app metadata

Key	Requirement	Description
`title`	**Mandatory**	The title will be displayed in the list of apps in the application manager.
`description`	**Mandatory**	The description will be displayed on the detail page of your app.
`authors`	**Mandatory**	Comma-separated list of authors.
`affiliations`	**Mandatory**	Single university/institution name or a comma-separted list of institutes in case of multiple affiliations.
`documentation_url`	**Mandatory**	The link to the online documentation of the app (e.g. on Read The Docs).
`video_installation_url`	**Mandatory**	A video of maximum length of 4 minutes showing how the app can be downloaded, installed, and put in working conditions.
`video_demonstration_url`	**Mandatory**	A video of maximum length of 4 minutes showing the app running under working conditions and being used, starting with a short explanation of the goal of the app and what it is trying to solve.
`external_url`	Optional	General homepage for your app.
`logo`	Optional	Url to a logo file (png or jpg).
`state`	Optional	One of - `registered` : lowest level - app may not yet be in a working state. Use this to secure a specific name. - `development` : app is under active development, expect the occasional bug. - `stable` : app can be used in production.
`industrial_collaboration`	Optional	If you have a closed-source app and you are an industrial partner of BIG-MAP.

Figure 5. Illustration of the table as seen on the README of the App Store repository with the valid keys for "metadata" part of the `apps.yaml` file.

As the registry evolves, the schema has been expanded and adapted regularly. We have recently added four new keys to the schema. The keys `video_installation_url` and `video_demonstration_url` are obviously related to the tutorial videos as discussed in section 2.2.2. The `affiliations` key helps keep track of whom to contact for future updates/news, as developers may switch jobs. Finally, the `industrial_collaboration` key keeps track of closed-source apps developed by BIG-MAP industrial partners.

As the number of apps registered on the App Store grew (from 5 in February 2021 to 29 in December 2023), more than 10 new categories were added to help with the description of the ever-increasing number of apps. This large number of new categories - some are related to machine learning, others to data analysis or virtual reality - reflects the great variety of newly registered apps (in terms of underlying technologies and use cases).

2.3 Additional changes

In addition to implementing the major technical changes listed above, we enforced that all registered apps adhere to the following requirements (as suggested by the European Commission):

1. The app's hosted source code repository (e.g., on GitHub) is publicly accessible.
2. The repository specifies the app's open-source license (e.g., in license.txt). Open-source licenses allow software to be freely used, modified, and shared. You can find a list of approved open-source licenses at this URL https://opensource.org/licenses.
3. The repository makes suitable acknowledgment to the funding by the European Commission using the following exact phrasing (e.g., in README.txt): "This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 957189. The project is part of BATTERY 2030+, the large-scale European research initiative for inventing the sustainable batteries of the future."
4. The repository provides sufficient technical documentation on how to install and run the software. This can be achieved via a README file, a Wiki page, or a software documentation hosting platform such as ReadTheDocs.

In the example of CleaseGUI, these requirements are fulfilled:

1. The source code is hosted on a public GitLab repository (https://gitlab.com/computationalmaterials/clease-gui)
2. The software license is Mozilla Public License Version 2.0
3. The README.md file contains the recommended acknowledgment (https://gitlab.com/computationalmaterials/clease-gui/-/blob/master/README.md)
4. The technical documentation, which is hosted on Read the Docs (https://clease-gui.readthedocs.io/en/latest), provides detailed information on how to install and run the software, as well as an example of application.

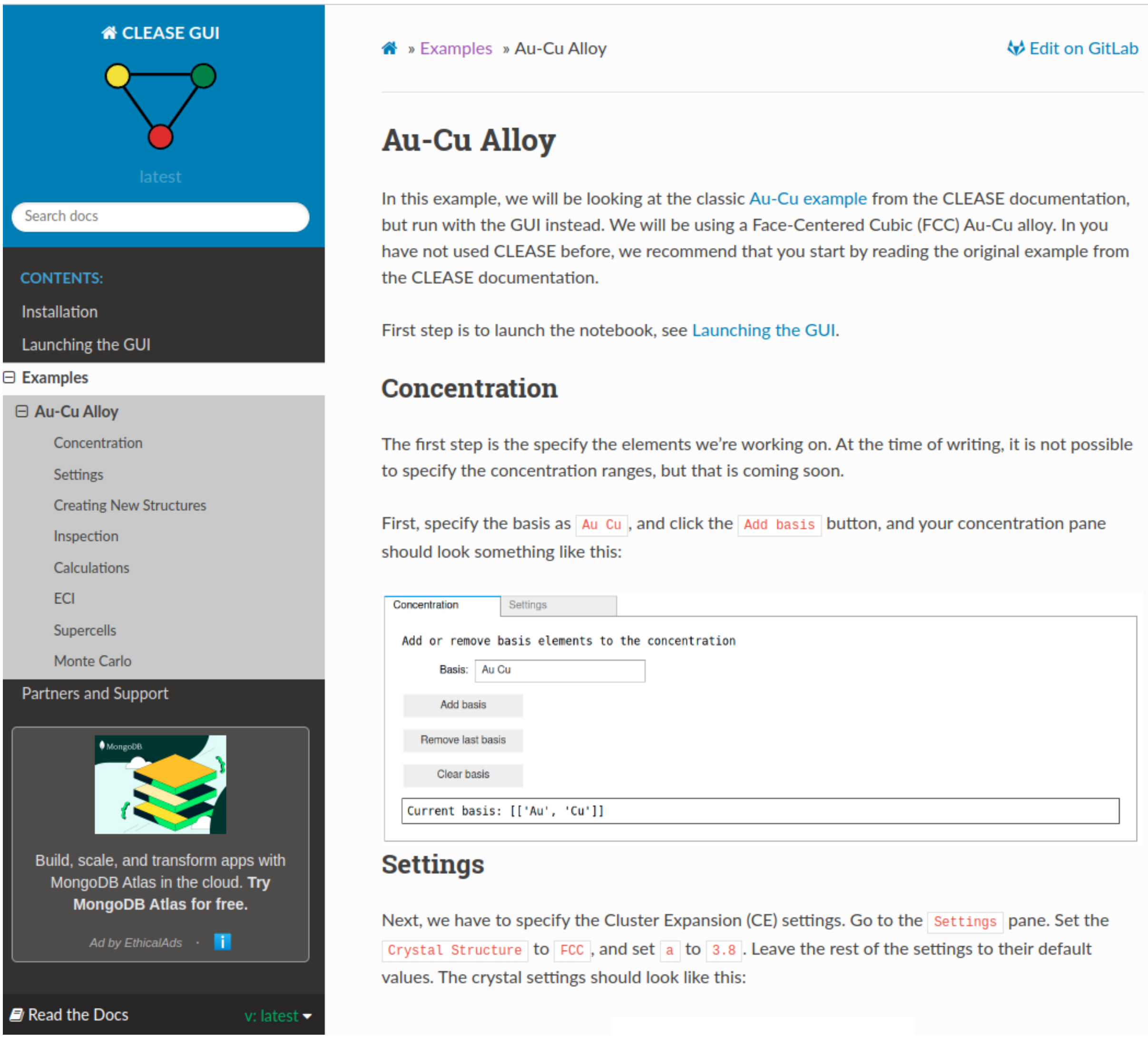

Figure 6. The homepage of CleaseGUI shows an example of how to run calculations using the graphical user interface with the Clease platform.

2.4 Deposition of apps

The process of adding new apps to the App Store is a straightforward process and the process is explained on the GitHub repository of the App Store. The authors/developers just need to fill up the details of their app in the `apps.yaml` file following the standardized schema described in section 2.2.3, and then open a pull request on the same repository. This entire process can be easily done in the web browser and usually takes only a few minutes.

3. App examples

As discussed in D9.1, the prototype of the App Store hosted 5 apps. In the intervening years, we have seen tremendous growth and have 6 times as many apps now. Of the 29 apps currently hosted on the App Store, 27 are completely open-source, and 24 have video tutorials. These apps tackle a wide array of topics ranging from automating electronic structure calculation workflows like Quantum ESPRESSO AiiDALab app, DFT-QE, DFT-VASP etc., automating experimental workflows like

Aurora AiiDALab, Self-driving labs, to data analysis like FullProfApp, MADAP etc., and data visualization like NMRium, Galvanicium etc.

3.1 Open-source apps

There are currently 27 open-source apps hosted on the App Store as shown in Figure 11. Figure 7 shows the homepage of the App Store which can be scrolled through to get a quick look at all the apps.

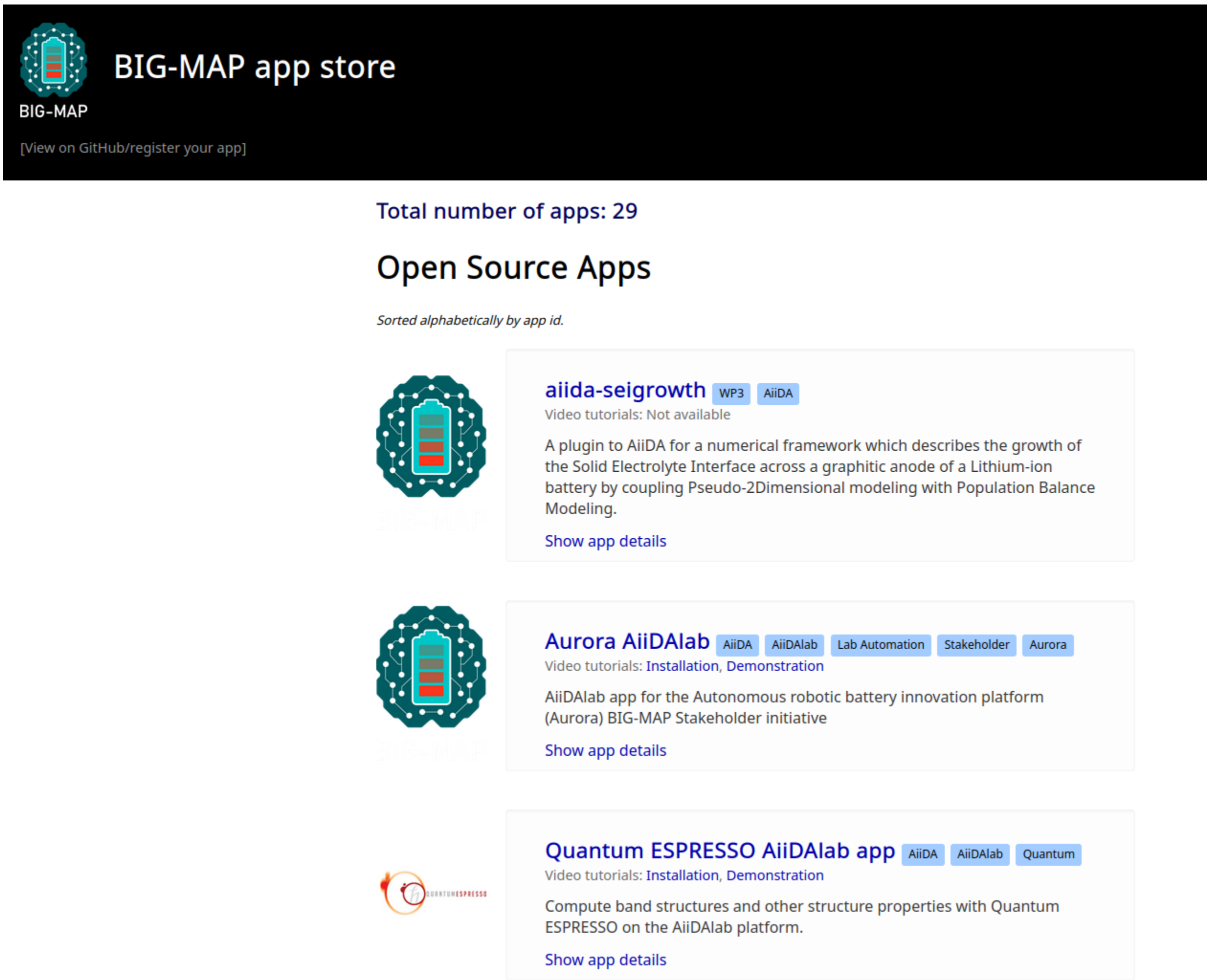

Figure 7. The homepage of the App Store, that can be scrolled through to get a brief overview of all the available apps. At a quick glance, it shows a brief description, the link to the GitHub repository, status of the app, and the links to the tutorial videos.

Clicking the app names, opens the page detailing the app as shown in Figure 8.

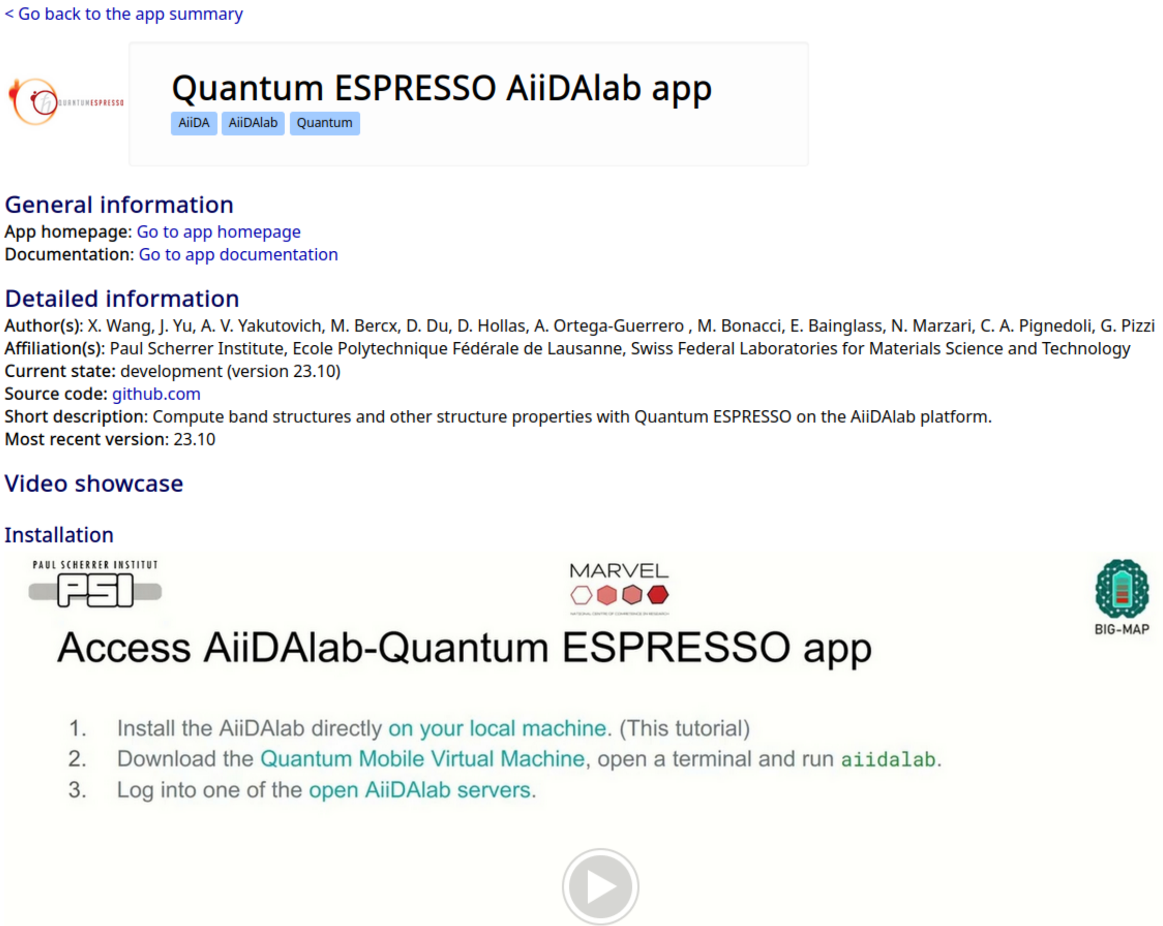

Figure 8. The "main" page of an app, that includes links to the app homepage, documentation, information on the app developers/authors and PI responsible, a short description, the current development status of the app, and video tutorials on installation and demonstration that can be played directly in the browser.

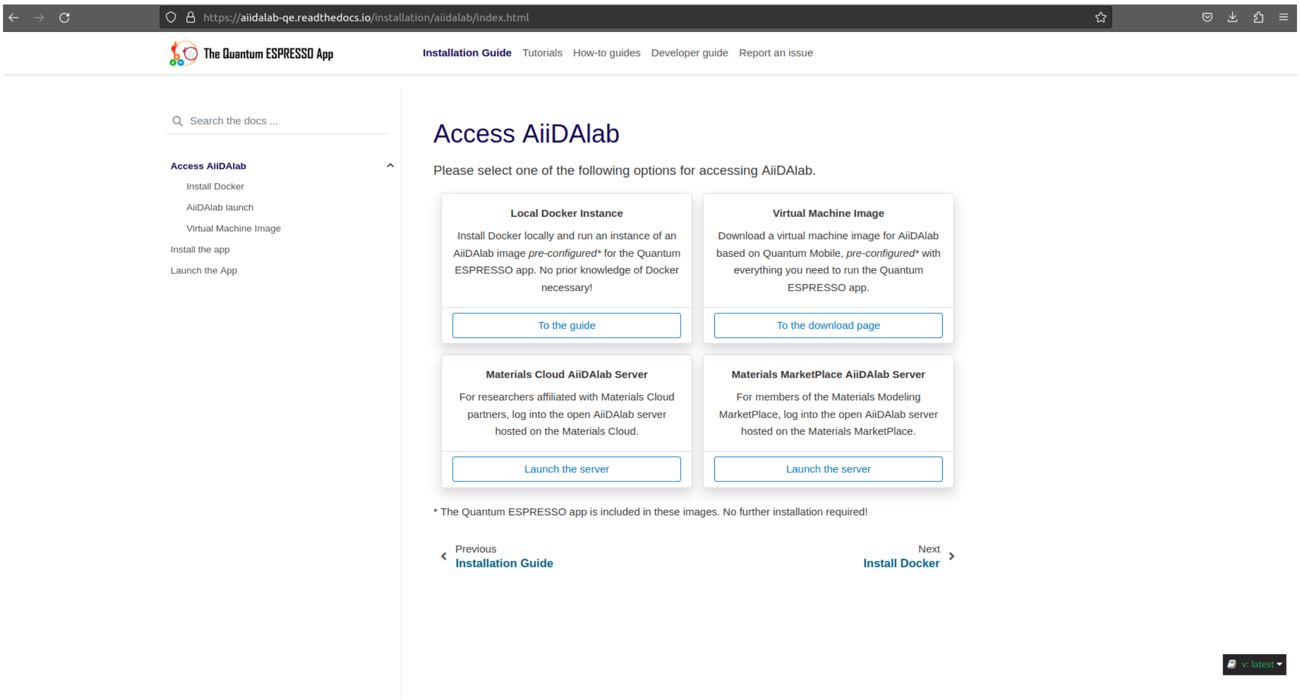

Figure 9. Following the link to app documentation shows the guide to install the apps, and includes other how-to guides as well.

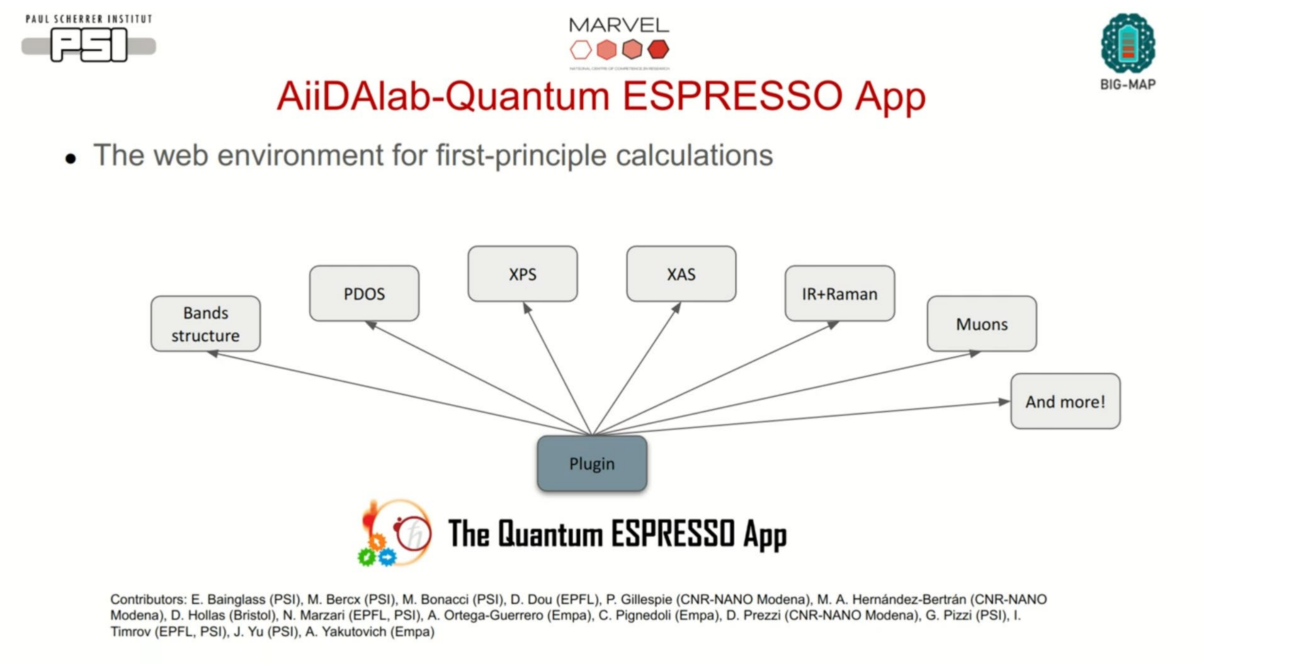

Figure 10. The working capabilities of the Quantum ESPRESSO AiiDALab app.

In Figure 11, we illustrate all the open-source apps that are currently hosted on the App Store.

aiida-seigrowth WP3 AiiDA
Video tutorials: Not available
A plugin to AiiDA for a numerical framework which describes the growth of the Solid Electrolyte Interface across a graphitic anode of a Lithium-ion battery by coupling Pseudo-2Dimensional modeling with Population Balance Modeling.
Show app details

Aurora AiiDAlab AiiDA AiiDAlab Lab Automation Stakeholder Aurora
Video tutorials: Installation, Demonstration
AiiDAlab app for the Autonomous robotic battery innovation platform (Aurora) BIG-MAP Stakeholder initiative
Show app details

Quantum ESPRESSO AiiDAlab app AiiDA AiiDAlab Quantum
Video tutorials: Installation, Demonstration
Compute band structures and other structure properties with Quantum ESPRESSO on the AiiDAlab platform.
Show app details

CCS (Curvature Constrained Splines) WP3 ASE
Video tutorials: Not available
CCS is a software tool to generate force fields in a flexible yet robust way.
Show app details

CLEASE GUI Data Analysis Utilities ASE
Video tutorials: Installation, Demonstration
A notebook-based graphical user interface for the Cluster Expansion code CLEASE. The app is only a GUI, and requires a working version of the CLEASE code. For more information on CLEASE, please refer to the CLEASE documentation (https://clease.readthedocs.io/).
Show app details

Electrolyte-Screening Quantum SimStack ASE
Video tutorials: Installation, Demonstration
In this workflow, we use the SimStack framework features to screening electrolytes systems using DFT calculation.
Show app details

EVA Data Visualization Data Analysis
Video tutorials: Installation, Demonstration
Electrochemistry Visualization Application is an application dedicated to electrochemical data analysis.
Show app details

FullProfAPP Data Analysis Utilities
Video tutorials: Installation, Demonstration
An PyQt5 based graphical user interface for the automated analysis of powder X-ray diffraction data based on the refinement engine FullProf.
Show app details

Galvanicium Data Visualization
Video tutorials: Installation, Demonstration
Galvanicium is a web application made to display and superimpose measurements from potentiostats and battery cyclers.
Show app details

HELAO Lab Automation Active Learning
Video tutorials: Installation, Demonstration
Small fastAPI based framework to deploy active learning and laboratory automation to a distributed fleet of research instruments. Contains many drivers and high level actions and datamanagement for a wide range of instruments.
Show app details

Datalab Data Management Data Analysis Stakeholder
Video tutorials: Installation, Demonstration
Datalab is a place to store your experimental data and the connections between them.
Show app details

densityNEB Utilities
Video tutorials: Installation, Demonstration
Code for calculating the path of least resistance between two points in a scalar field using nudged elastic band (NEB).
Show app details

DFT-QE Quantum SimStack ASE
Video tutorials: Installation, Demonstration
The DFT-QE WaNo implements various methods available within the Quantum Espresso code.
Show app details

DFT-Surface Quantum SimStack
Video tutorials: Not available
This workflow uses the SimStack framework features to perform as an option a single shot DFT calculation of molecules absorbing on a surface.
Show app details

DFT-VASP Quantum SimStack
Video tutorials: Installation, Demonstration
The DFT-VASP WaNo implements a wide range of methods available within the VASP code.
Show app details

HierCVAE Generative Model Inverse Design MLMD WP11
Video tutorials: Installation, Demonstration
Hierarchical Conditional Variational Autoencoder. A BioLib app for inverse design of molecules given the electron band gap.
Show app details

MADAP Data Analysis Data Visualization Utilities
Video tutorials: Installation, Demonstration
Modular and Autonomous Data Analysis Platform (MADAP) is a well-documented python package which can be used for electrochmeical data analysis for three different classes: Impedance, Arrhenius and Voltammetry
Show app details

NMRium Data Visualization Data Analysis
Video tutorials: Installation, Demonstration
NMRium is a tool for visualizing 1D and 2D NMR spectra that are stored in JCAMP-DX format directly in your web browser.
Show app details

Onterface Data Visualization Data Management Utilities WP7 Stakeholder
Video tutorials: Installation, Demonstration
Semantic web platform demonstrator coupling knowledge graphs with simulation and optimization workflows to enable on-click end-user solutions
Show app details

OPTIMADE Web Client Materials Cloud Utilities
Video tutorials: Demonstration
Graphical client hosted on Materials Cloud to search databases that implement the OPTIMADE API (https://www.optimade.org/).
Show app details

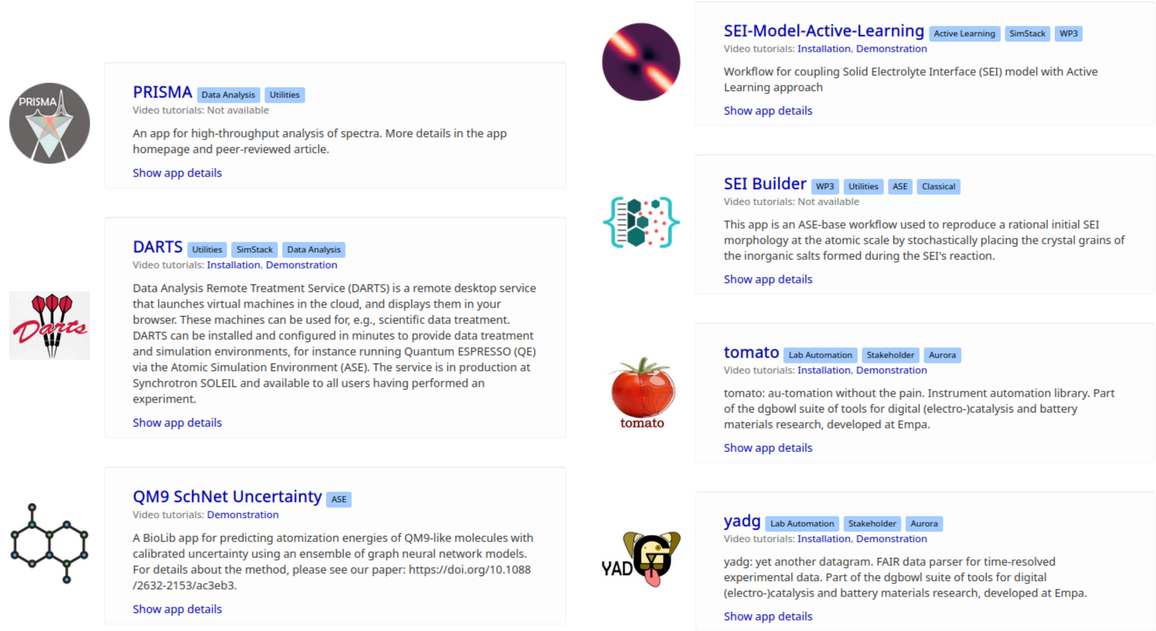

Figure 11. The full list of all the open-source apps currently hosted on the App Store.

3.2 Industrial partners apps

These apps use proprietary methodology, but the usage of the apps is available publicly. There are two such apps hosted on the App Store as shown in Figure 12, and both are developed by the industrial partners of BIG-MAP.

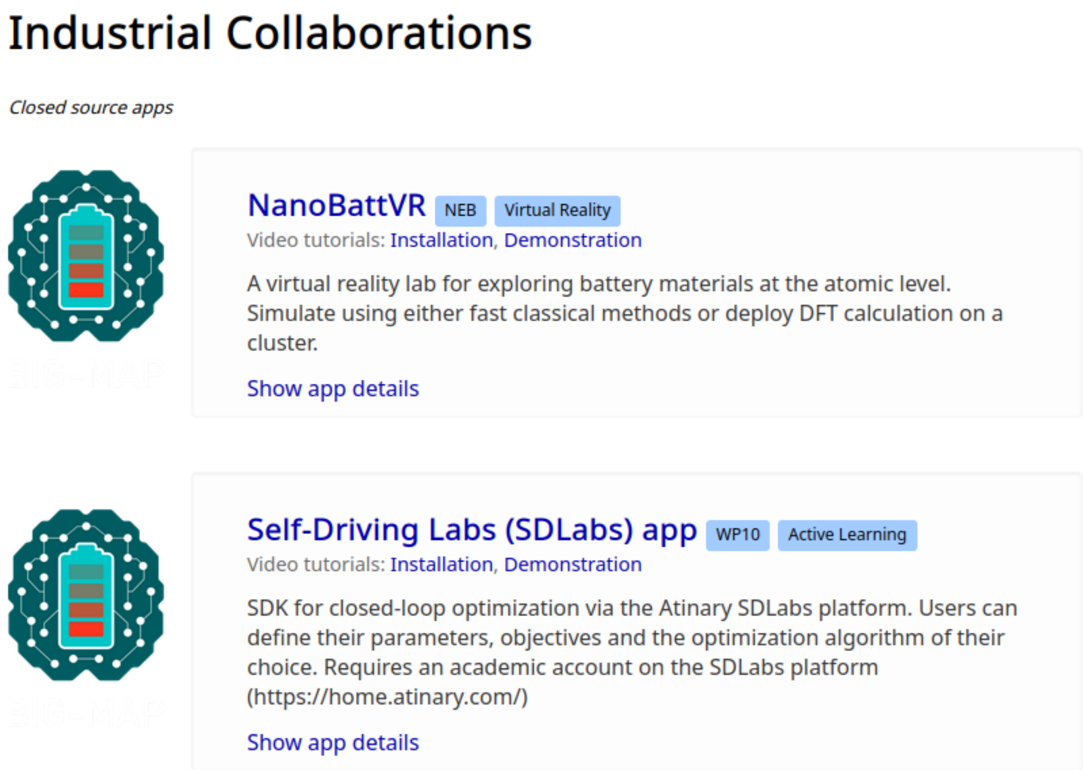

Figure 12. The apps developed by industrial partners of BIG-MAP.

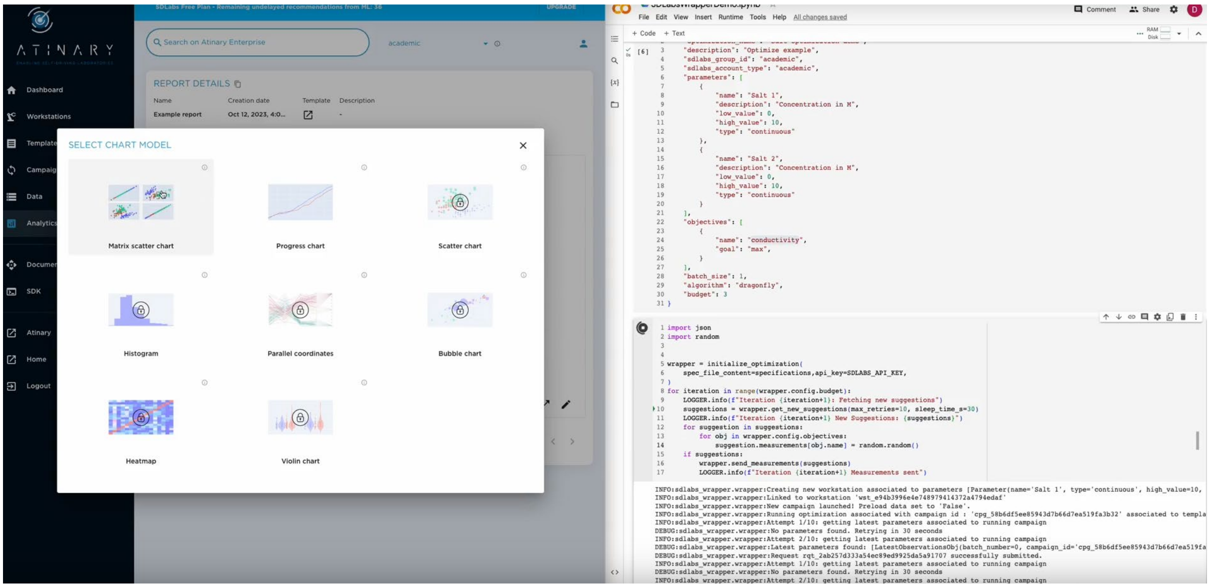

Figure 13. A demonstration of SD labs wrapper app, showing how a simple pythonic code can be used to drive the powerful SDLabs platform to run optimization workflow for cell geometry formulation.

4. Conclusions

In conclusion, the BIG-MAP App Store (https://big-map.github.io/big-map-registry/) has now made available to the battery community worldwide 29 apps (27 open-source from BIG-MAP, 2 from industrial partners) offering many key capabilities – from automated first-principles DFT calculations (e.g., with Quantum ESPRESSO or VASP) to cluster-expansion methods for the calculation of temperature-composition phase diagrams, to experimental apps for electrochemical data analysis or for the autonomous driving of robotic experiments.

The BIG-MAP App Store thus serves as a forward-looking solution to the challenge of disseminating computational tools and methodologies in the field of battery research. The curation process plays a central role in ensuring the success and effectiveness of the App Store, for which we host monthly meetings attended by WP9 members, where all the app developers/authors are invited to receive valuable feedback on how to further improve their apps and how the App Store can streamline the access to these apps.

The major technical overhauls demonstrate a commitment to enhancing user experience and public accessibility. The inclusion of tutorial videos for each app not only showcases their capabilities but also facilitates a user-friendly understanding of installation and the usage procedures. The expansion of the metadata schema with new fields and categories to accommodate the growing number and diverse nature of registered apps reflects the evolution of the App Store from a prototype to a rich repository covering a wide spectrum of battery research topics, from theoretical calculations to experimental data analysis and visualization. The adherence to European Commission requirements of explicit open-source licensing, acknowledgment of funding and the provision of robust technical documentation, ensures transparency, accessibility and compliance with regulatory standards.

In summary, the BIG-MAP App Store is not simply a repository but an important platform to foster collaboration, facilitate knowledge dissemination and promote transparency in the field of battery research. Through continuous curation efforts, technical enhancements and a commitment to openness, the App Store has become the go-to place for all the research within BIG-MAP.