Paulownia Circular-Economy Dashboard

This repository contains a prototype dashboard for modelling the circular economy of *Paulownia* forestry projects. It is implemented in Python using Streamlit and Plotly. The dashboard allows you to explore agro-forestry growth scenarios, biomass flows, logistics and processing chains, and end-of-life soil carbon projects. See `docs/USER_GUIDE.md` for a walkthrough.

The code is designed around a deterministic core where all calculations are performed off-line using the parameters you provide. No external APIs are required. A nested set of Pydantic models describe the various stages (agro, logistics, extraction, substrate production, plate manufacturing and end-of-life), and these models are serialisable to JSON for scenario sharing.

Quick start

```
# create a virtual environment and install dependencies
python -m venv .venv
source .venv/bin/activate
pip install -r requirements.txt

# run unit tests
# pytest -q

# launch the dashboard
streamlit run app.py
```

Project structure

```
paulownia_dash/
├── app.py                        # Streamlit entry point
├── core/                         # Deterministic models and computations
│   ├── __init__.py
│   ├── params.py                # Pydantic data models
│   ├── sim.py                   # Agro-forestry simulator
│   ├── sim_extraction.py        # Logistics, extraction, substrate and plates
│   ├── sim_eol.py               # End-of-life soil carbon module
│   ├── aggregate.py             # Merge dataframes and compute KPIs
│   ├── economics.py             # NPV/IRR utilities
│   ├── plots.py                 # Plotly figure builders
│   └── utils.py                 # Miscellaneous helpers
├── pages/                        # Individual Streamlit pages
│   ├── 1_🌳_Scenario_Inputs.py
│   └── 2_📈_Results_Timeseries.py
│   │__...
├── docs/                         # User and developer documentation
│   ├── USER_GUIDE.md
│   ├── DEV_GUIDE.md
```

```
│   ├── ROADMAP.md
│   └── QA_REPORT.md
├── assets/                     # Theming and presets
│   └── theme.json
├── tests/                      # PyTest unit and smoke tests
│   ├── test_core.py
│   ├── test_aggregate.py
│   ├── test_pages_smoke.py
│   └── test_downloads.py
├── .streamlit/
│   └── config.toml             # Streamlit configuration
├── scripts/
│   └── make_zip.py             # Utility to build release zip
├── requirements.txt            # Exact dependency versions
└── Makefile                    # Convenience commands
```

Extending the model

To add a new stage or expand an existing one, follow these guidelines:

- Define new parameters in `core/params.py` with sensible defaults and type hints.
- Write pure functions in a new module under `core/` that take these parameters and return `pandas.DataFrame` objects describing yearly results.
- Update `core/aggregate.py` to merge your new dataframe into the main `df_joined`. Be sure to recalculate KPIs as needed.
- Expose sliders and inputs in a new or existing page under `pages/`, using `st.form` to group inputs. Cache the simulation results with `st.cache_data` keyed on the scenario JSON.
- Add tests under `tests/` to validate your computations.

Licence

This prototype is released under the MIT licence. See `LICENSE` for details.