Tech Stack

- Backend — Tech Stack
- Frontend - Tech Stack
- Tech Stack Overview
- Auxillary

Backend — Tech Stack

Repository: `finmars-platform/finmars-core` and `finmars-platform/finmars-workflow`

Overview

This document lists the technologies, libraries, and approaches used in the core and workflow backend.

Language

- **Python** (primary language)

Frameworks & Runtime

- **Django** (project structure with `manage.py`, templates)
- **Django REST Framework (DRF)** (API layer)
- **Gunicorn** (WSGI/ASGI server for running Django in production)

Dependency Management

- **Poetry** (`pyproject.toml` present)
- **pip/requirements** files for environments: `requirements.txt`, `requirements-dev.txt`, `requirements-test.txt`

Containerization & Local Orchestration

- **Docker** (`Dockerfile`)
- **Docker Compose** (`docker-compose.yml`)
- **Makefile** for common dev ops (`make migrate`, `make up`, etc.)

CI/CD

- **GitHub Actions** for build/test/deploy pipelines
- Builds use the provided **Dockerfile**

Scripts & Tooling

- Shell scripts for setup and maintenance: `generate_env.sh`, `migrate.sh`, `import-sql.sh`, `graph.sh`

Configuration

- `.env.sample` for environment variables example
- Standard Django settings (inside the project; details depend on env files)

Database

- **PostgreSQL 15**

Storage (one of options)

- **S3 / Minio/ Azure Storages**
- **Local File Storage**
- **NFS Storage**

Caching

- **Redis** (used for caching)

Message Queue & Background Tasks

- **RabbitMQ** (used for task queues)
- **Celery** (used for background and scheduled tasks)

Authentication

- **Keycloak** (mandatory integration for authentication)

Testing

- Test requirements listed in `requirements-test.txt` (exact test tools depend on that file, e.g., pytest)

Libraries

The full list of libraries is declared in `requirements.txt` and `pyproject.toml`. These include (but are not limited to):

```
aiohappyeyeballs==2.6.1
aiohttp==3.11.18
aiosignal==1.3.2
alabaster==1.0.0
amqp==5.3.1
asgiref==3.8.1
async-timeout==5.0.1
attrs==25.3.0
azure-core==1.31.0
azure-storage-blob==12.14.1
babel==2.17.0
bcrypt==4.3.0
beautifulsoup4==4.13.3
billiard==4.2.1
boto3==1.37.17
botocore==1.37.38
celery==5.4.0
certifi==2025.1.31
cffi==1.17.1
charset-normalizer==3.4.1
click==8.1.8
click-didyoumean==0.3.1
click-plugins==1.1.1
click-repl==0.3.0
coreapi==2.3.3
coreschema==0.0.4
```

```
cron-descriptor==1.4.5
croniter==6.0.0
cryptography==44.0.2
deepdiff==8.4.2
Django==4.2.22
django-celery-beat==2.7.0
django-celery-results==2.5.1
django-cors-headers==3.14.0
django-cprofile-middleware==1.0.5
django-crispy-forms==1.14.0
django-cuser==2017.3.16
django-filter==21.1
django-js-asset==2.2.0
django-modeltranslation==0.18.11
django-mptt==0.16.0
django-redis==5.4.0
django-rest-swagger==2.2.0
django-storages==1.14.5
django-timezone-field==7.1
django-tinymce==4.1.0
django_debug_toolbar==3.8.1
djangorestframework==3.15.2
djangorestframework-simplejwt==5.5.0
docutils==0.21.2
drf-yasg==1.21.10
ecdsa==0.19.1
et_xmlfile==2.0.0
factory_boy==3.3.3
Faker==37.1.0
filtration==2.3.0
flower==2.0.1
frozenlist==1.6.0
geoip2==5.0.1
gunicorn==23.0.0
humanize==4.12.2
idna==3.10
imagesize==1.4.1
inflection==0.5.1
ipcalc==1.99.0
```

```
isodate==0.7.2
itypes==1.2.0
Jinja2==3.1.6
jmespath==1.0.1
kombu==5.5.3
MarkupSafe==3.0.2
maxminddb==2.6.3
memory-profiler==0.61.0
msrest==0.7.1
multidict==6.4.3
numpy==2.2.4
oauthlib==3.2.2
openapi-codec==1.3.2
openpyxl==3.0.10
orderly-set==5.4.0
packaging==25.0
pandas==2.2.3
paramiko==3.5.1
pbr==6.1.1
pexpect==4.8.0
pillow==11.3.0
pprintpp==0.4.0
prometheus_client==0.21.1
prompt_toolkit==3.0.51
propcache==0.3.1
psutil==7.0.0
psycopg2-binary==2.9.10
ptyprocess==0.7.0
pyasn1==0.4.8
pycparser==2.22
pycryptodome==3.19.1
Pygments==2.19.1
PyJWT==2.6.0
Pympler==1.1
PyNaCl==1.5.0
pyOpenSSL==25.0.0
pyotp==2.7.0
pyparsing==3.0.9
python-crontab==3.2.0
```

```
python-dateutil==2.8.2
python-jose==3.4.0
pytz==2025.1
PyYAML==6.0.2
QuantLib==1.37
redis==5.2.1
requests==2.32.4
requests-oauthlib==2.0.0
rsa==4.9.1
ruff==0.11.6
s3transfer==0.11.5
sentry-sdk==2.26.1
simplejson==3.20.1
six==1.17.0
snowballstemmer==2.2.0
soupsieve==2.7
Sphinx==8.1.3
sphinxcontrib-apidoc==0.5.0
sphinxcontrib-applehelp==2.0.0
sphinxcontrib-devhelp==2.0.0
sphinxcontrib-django==2.5
sphinxcontrib-htmlhelp==2.1.0
sphinxcontrib-jsmath==1.0.1
sphinxcontrib-qthelp==2.0.0
sphinxcontrib-serializinghtml==2.0.0
sqlparse==0.5.3
suds==1.2.0
tomli==2.2.1
tornado==6.5.0
typing_extensions==4.13.2
tzdata==2025.2
uritemplate==4.1.1
urllib3==2.5.0
vine==5.1.0
wcwidth==0.2.13
whitenoise==6.9.0
yarl==1.20.0
```

License

- **EUPL-1.2** (repository license)

Frontend - Tech Stack

Repository: `finmars-platform/finmars-vue-portal` and `finmars-platform/finmars-portal` and `finmars-platform/finmars-workflow-portal`

Overview

This document describes the technology stack used in the Vue Portal frontend.

Language

- **JavaScript / TypeScript**

Framework

- **Vue 3** (main frontend framework)
- Angular.js (for `finmars-platform/finmars-portal`)

Build Tool

- **Vite**

State Management

- **Pinia**

Router

- **Vue Router**

UI / Styling

- **CSS / SCSS**
- **TailwindCSS** or UI component library (Naive UI / Element Plus / Vuetify)

API Layer

- **HTTP Client**: likely **Axios**
- Backend: **Django REST Framework** (from core), communication over REST/JSON

Authentication

- **Keycloak** (mandatory integration). Client library: `keycloak-js`

Environment & Config

- `.env` files for environment variables (e.g., `VITE_*`) — exact names to be confirmed

Testing

- **E2E Testing**: Cypress / Playwright (separate repository)

Linting & Formatting

- **ESLint**, **Prettier** — rules and configs

CI/CD

- **GitHub Actions** (build and deploy). Build process based on **Dockerfile**

Containerization

- **Docker** (`Dockerfile`)
- **Docker Compose** for local development — to be confirmed

Observability

- Logging and monitoring for the frontend — Sentry

License

- **EUPL-1.2** (repository license)

finmars-vue-portal package.json


```
{
  "private": true,
  "scripts": {
    "build": "nuxt build ",
    "build:web-component": "vite build",
    "dev": "nuxt --rootDir=./v/ dev -p=3000",
    "dev-local": "NUXT_APP_BUILD_ASSETS_DIR=/realm00000/v/_nuxt/ API_HOST=http://0.0.0.0:8000
AUTH_HOST=http://0.0.0.0:8083/authorizer nuxt --rootDir=./v/ dev --host=0.0.0.0",
    "start": "PORT=8080 node .output/server/index.mjs",
    "generate": "nuxt generate",
    "preview": "nuxt preview",
    "test": "vitest",
    "coverage": "vitest run --coverage"
  },
  "author": "Finmars SCSA",
  "license": "EUPL-1.2",
  "devDependencies": {
    "@nuxt/test-utils": "^3.12.1",
    "@nuxtjs/tailwindcss": "^6.12.1",
    "@types/sanitize-html": "^2.13.0",
    "@vitest/coverage-v8": "3.2.4",
    "@vue/test-utils": "^2.4.5",
    "consola": "^3.2.0",
    "eslint": "^8.24.0",
    "eslint-plugin-vue": "^9.6.0",
```

```
    "happy-dom": "^17.0.0",
    "nuxt": "^3.11.0",
    "postcss-mixins": "^10.0.1",
    "prettier": "3.3.3",
    "sass": "1.85.1",
    "unplugin-auto-import": "^0.16.6",
    "unplugin-vue-components": "^0.25.1",
    "vite-plugin-commonjs": "^0.10.1",
    "vitest": "3.2.4"
  },
  "dependencies": {
    "@chenfengyuan/vue-qrcode": "^2.0.0",
    "@finmars/ui": "1.0.87",
    "@formbricks/js": "1.1.2",
    "@kyvg/vue3-notification": "^2.4.1",
    "@pinia/nuxt": "^0.5.1",
    "@vue/web-component-wrapper": "^1.3.0",
    "ace-builds": "^1.36.2",
    "chart.js": "^3.9.1",
    "click-outside-vue3": "^4.0.1",
    "dayjs": "^1.11.7",
    "esbuild": "^0.25.5",
    "eslint-config-prettier": "^9.1.0",
    "eslint-plugin-prettier": "^5.2.1",
    "floating-vue": "^5.2.2",
    "grid-layout-plus": "^1.0.6",
    "jquery": "^3.7.1",
    "jsondiffpatch": "^0.6.0",
    "jstree": "^3.3.17",
    "keycloak-js": "^21.0.1",
    "lodash": "^4.17.21",
    "pdfjs-dist": "^4.0.269",
    "pinia": "^2.0.24",
    "qrcode": "^1.5.1",
    "sanitize-html": "^2.13.1",
    "swiper": "^10.0.4",
    "vue": "^3.4.23",
    "vue-matomo": "^4.2.0",
    "vue3-ace-editor": "^2.2.2",
    "vuedraggable": "^4.1.0"
```

```
□}
}
```

Tech Stack Overview

This document provides a high-level overview of the technologies used across the Finmars Platform.

Backend (Core & Workflow)

- **Language**: Python
- **Framework**: Django, Django REST Framework (DRF)
- **Background Tasks**: Celery
- **Database**: PostgreSQL 15
- **Caching**: Redis
- **Message Queue**: RabbitMQ
- **Authentication**: Keycloak
- **Runtime**: Gunicorn
- **Containerization**: Docker, Docker Compose
- **CI/CD**: GitHub Actions

Frontend (Vue Portal and Portal)

- **Framework**: Vue 3 / Angular.js
- **Build Tool**: Vite / Webpack (to confirm)
- **State Management**: Pinia / Vuex (to confirm)
- **Routing**: Vue Router
- **Styling/UI**: CSS/SCSS, TailwindCSS or UI library (to confirm)
- **HTTP Client**: Axios (likely)
- **Authentication**: Keycloak integration
- **Containerization**: Docker
- **CI/CD**: GitHub Actions

Shared Practices

- **Environment Variables** via `.env` files
- **Dockerfile-based builds**

- **Automated testing** (pytest for backend, Jest/Vitest/Cypress for frontend — to confirm)

Cloud

- **Kubernetes**
- **AWS S3 / Minio**
- **AWS IAM**
- **Hypervisors (esxi, xcp-ng)**
- **Hashicorp Vault**
- **Pritunl VPN**

Other

- **Ionic Framework**
- **git (github, gitlab)**
- **Wordpress**
- **Storybook**
- **Cypress**
- **Jira/Confluence**
- **Various integrations (bloomberg, exante, eod, etc.)**

License

- **EUPL-1.2**

Auxillary

Cloud, Community Edition

Overview

This document describes the technology stack used in the Cloud / Community Edition installation

Language

- **Bash**
- **Python**

Technologies

- **Kubernetes**
- **SSL**
- **Nginx**
- **HTTPS**
- **Encrypted Disks**
- **Docker**
- **VPN (pritunl)**
- **Router (pfsense)**
- **AWS**
- **Azure**
- **Various Cloud Providers with Linux Support**
- **Ionic Framework**