

AerialyticTechnicalAssignment Project File Summary

This document provides a high-level summary of each file and directory in the project, excluding the venv directory.

Root Directory

- Aerialytic Technical Assignment (1).pdf: The original technical assignment document outlining the project requirements.
- docker-compose.yml: Defines and manages multi-container Docker applications for local development, including backend and frontend services.
- README.md: Project overview and instructions.

backend/

- db.sqlite3: SQLite database file for local development.
- Dockerfile: Instructions to build the backend Docker image.
- manage.py: Django's command-line utility for administrative tasks.
- pyproject.toml: Python project metadata and dependencies.
- README.md: Backend-specific documentation.
- requirements.txt: Python dependencies for the backend.
- uv.lock: Lock file for Python dependencies.
- solar/: Django app containing the core backend logic.
 - **init.py**: Marks the directory as a Python package.
 - admin.py: Django admin interface configuration for the solar app.
 - apps.py: App configuration for Django.
 - migrations/: Database schema migrations for the solar app.
 - * **init.py**: Marks the migrations directory as a Python package.
 - * 0001_initial.py: Initial database schema for the solar app.
 - models.py: Database models for storing solar calculation data.
 - solar_calculator.py: Core logic for solar panel calculations and optimization.
 - tests.py: Unit and API tests for the solar app.
 - urls.py: URL routing for the solar app's API endpoints.
 - views.py: API endpoint implementations for calculations and health checks.
- solar_backend/: Django project configuration.
 - **init.py**: Marks the directory as a Python package.
 - asgi.py: ASGI entrypoint for asynchronous server support.
 - settings.py: Main Django settings file.
 - urls.py: Project-level URL routing.
 - wsgi.py: WSGI entrypoint for web server support.
- venv/: (ignored)

frontend/

- Dockerfile: Instructions to build the frontend Docker image.
- nginx.conf: Nginx configuration for serving the frontend and proxying API requests.
- package.json: Node.js dependencies and scripts for the frontend.
- package-lock.json: Lock file for Node.js dependencies.
- README.md: Frontend-specific documentation.
- tsconfig.json: TypeScript configuration for the frontend.
- public/: Static assets and the main HTML file for the React app.
 - favicon.ico, index.html, logo192.png, logo512.png, manifest.json, robots.txt: Standard React static files and metadata.
- src/: Source code for the React frontend.
 - App.css, App.tsx, App.test.tsx, index.css, index.tsx, logo.svg, react-app-env.d.ts, reportWebVitals.ts, setupTests.ts: Main React app files, entrypoints, and test setup.
 - components/: React components for the UI.
 - * CoordinateForm.tsx: Form for entering coordinates.
 - * LocationMap.tsx: Map component for selecting a location.
 - * ResultsDisplay.tsx: Displays calculation results.
 - * SolarPanelCalculator.tsx: Main calculator logic and UI.
 - types/: TypeScript type definitions.
 - * solar.ts: Types for solar calculation data.

helm/

- Chart.yaml: Helm chart metadata for Kubernetes deployments.
- README.md: Helm chart usage instructions.
- values.yaml: Default configuration values for the Helm chart.

k8s/

- backend-deployment.yaml: Kubernetes deployment and service for the backend.
- backend-hpa.yaml: Horizontal Pod Autoscaler configuration for backend autoscaling.
- configmap.yaml: Configuration data for Kubernetes resources.
- frontend-deployment.yaml: Kubernetes deployment and service for the frontend.
- frontend-hpa.yaml: Horizontal Pod Autoscaler for the frontend.
- ingress.yaml: Ingress resource for external access to services.
- logging-efk.yaml: EFK (Elasticsearch, Fluentd, Kibana) stack for logging.
- monitoring-prometheus.yaml: Prometheus monitoring stack configuration.
- postgres-deployment.yaml: Kubernetes deployment and service for a PostgreSQL database.
- README.md: Kubernetes deployment instructions and documentation.