

# Proj 1 - Week 1 | Earth Light Curve

## Objective

Lay the foundation for the Earth Light Curve app: define the problem and user stories, select datasets/AOIs, lock the tech stack, wireframe the UI, and ship a clean repo with CI, tests, and a minimal working scaffold (map + chart shells).

## Outcomes (by end of Week 1)

- Public repo with a running scaffold (Map + empty chart panel) and green CI.
- Clear **user stories**, **acceptance criteria**, **success metrics** for the month.
- Chosen **sensor & STAC source**, **AOIs**, and a short **data spec** (bands, masks).
- Low-fi **wireframes** (key screens + interaction notes).
- A concise **Tech Decisions** doc (1-2 pages) with a data-flow diagram.
- Basic **tests** (2+ unit, 1+ E2E), URL-state wired for AOI/dates.

## Tasks & Guidelines

### 1) Problem framing & user stories

#### Tasks

- Write a 1-paragraph problem statement: “*Enable a user to draw/select a region and view a de-clouded NDVI/reflectance time series with basic change detection.*”
- Draft **3-5 user stories**, e.g.:
  - *As a user, I can draw a polygon to get a time series for that area.*
  - *As an analyst, I can copy a shareable URL (AOI + dates + filters).*
  - *As a reviewer, I can export the series as CSV/PNG.*
- Define **success metrics** (e.g., initial load < 2.5s, series response < 2s, CWV “good”).

**Deliverable:** docs/product/user-stories.md ( $\leq 1$  page).

### 2) Data selection & spec

#### Tasks

- Choose **sensor**: recommended **Sentinel-2 L2A**.
- Define initial **index**:  $NDVI = (NIR - Red)/(NIR + Red)$ .
  - S2 bands: Red = B4, NIR = B8 (note B8A as an alternative).
- Decide **cloud strategy** for Week 2 and document now:
  - Option A: product QA layer (e.g., scene classification / cloud mask).
  - Option B: simple band thresholds + morphological clean-up (fallback).
- Pick **2-3 AOIs** with varied land cover (forest, urban, cropland). For each:
  - Name, rationale, bounding box (lat/lon), small screenshot.
- Pick a **public STAC API** (base URL, key query params: bbox, datetime, cloud %).
- Note **projection & tiling** choices (Web Mercator for map UI; source UTM kept).

**Deliverable:** docs/data/data-spec.md (sensor, bands, masks, AOIs, STAC query examples).

### 3) Architecture & stack decisions

#### Tasks

- Lock **stack**: React + TypeScript; MapLibre for map; lightweight chart lib (Recharts/ECharts); Vite or Next.js (choose + justify); state (Zustand/Context); testing (Jest + RTL + Playwright).
- Draft a **data-flow diagram**: ROI → STAC search → items → sampling/composite → series → chart.
- Define **URL-state schema** (AOI, date range, cloud %, band combo, index).
- Set **performance budgets** (initial bundle  $\leq$  300 KB; keep map/style tiles external).
- Set **a11y goals** (keyboard order, ARIA landmarks/roles, visible focus).

**Deliverable:** docs/tech/decisions.md + docs/tech/data-flow.png.

### 4) Repo, environment & CI

#### Tasks

Create a public GitHub repo with structure:

- /src
- /components
- /features/map
- /features/series
- /lib
- /public
- /tests
- /e2e
- /docs

Add ESLint, Prettier, strict TS; .nvmrc for Node version; lockfile committed.

- NPM scripts: dev, build, preview, test, e2e, lint, format, typecheck.
- **GitHub Actions**: PR → install/lint/unit/E2E/build; main → deploy (Vercel/Netlify/GH Pages).
- Add README.md, CONTRIBUTING.md, PR template, **CODEOWNERS** (you).

**Deliverable:** Repo link; first PR shows green CI + preview deploy.

### 5) UI wireframes (low-fi) & interactions

#### Tasks

- Sketch **3 screens** (hand-drawn or Figma low-fi):
  1. **Home**: map + sidebar (AOI tools), date range, cloud slider.
  2. **Series view**: chart panel with tooltips, smoothing toggle, export.

- 3. **Empty/error state:** helpful copy + “Use sample AOI” CTA.
- Annotate **interactions:** ROI draw/modify, keyboard shortcuts, loading states.
- Add **a11y notes:** focus order, roles, escape routes.

**Deliverable:** docs/ux/wireframes/\*.png + docs/ux/notes.md ( $\leq 1$  page).

## 6) Minimal working scaffold

### Tasks

- Render **MapLibre** with a neutral basemap; enable **ROI draw** (point + polygon). Log coordinates to console for now.
- Add **chart placeholder** (static series) to validate layout & resize.
- Implement **URL state** read/write for AOI + dates.
- Add **tests:**
  - Unit: map and chart components mount; URL parser/serializer.
  - E2E: open app → draw polygon → URL reflects AOI → refresh restores state.

**Deliverable:** App runs locally and deploys on PR; README shows a short GIF.

## Acceptance Criteria (Week 1)

- **Docs:** user-stories.md, data-spec.md, decisions.md, wireframes + notes.
- **Scaffold:** Map + chart shells render; ROI draw logs lat/lon; URL state works.
- **Repo hygiene:** ESLint/Prettier/TS strict; CI passing; deploy preview link active.
- **Tests:**  $\geq 2$  unit +  $\geq 1$  E2E passing on CI.
- **Performance baseline:** initial bundle  $\leq 300$  KB (excluding map tiles).
- **Accessibility baseline:** keyboard focus traverses map  $\leftrightarrow$  sidebar  $\leftrightarrow$  chart with visible focus.

## Submission (end of Week 1)

- GitHub repo (public) + deployed preview link.
- A **90-sec Loom** (or similar) walking through: problem, wireframes, scaffold, next-week plan.
- A 5-7 sentence **README intro** summarizing Week 1 status.

## Stretch (optional)

- **Sample AOI** button that injects a known polygon and updates URL.
- **Mock STAC JSON** + a thin fetch wrapper to stabilize next week’s work.
- **Clipboard:** “Copy shareable link” utility with toast confirmation.
- **Error boundaries** and friendly empty states.

## Skills you’ll add (toward your CV after 4 projects)

- Front-end: **React + TypeScript**, component patterns, state management, URL-state design.

- Geospatial UI: **MapLibre**, ROI drawing, tiling/projection awareness (Web Mercator vs UTM).
- Data plumbing: **STAC** querying patterns, COG preview basics, provenance & QA flags.
- Time-series UX: chart scaffolding, tooltips, export flows, future smoothing/change-point hooks.
- Engineering quality: **Jest/RTL/Playwright**, CI/CD with preview deploys, **performance budgets**, **a11y** foundations.
- Reproducibility: versioned decisions, lockfiles, Node version pinning, repo hygiene.

## Study topics (EO & Astro) relevant to this project

We discussed some of these topics today but I would like you to read more and be prepared to discuss them further.

### Remote Sensing (EO)

- Sentinel-2 L2A product structure; bands (B2-B8/B8A/B11/B12) & resolutions.
- NDVI and index basics; interpreting vegetation dynamics.
- Cloud masking: scene classification (SCL), QA layers, simple thresholding; strengths/limits.
- COG, STAC concepts, and why they matter for scalable access.
- Temporal compositing, resampling, and BRDF/illumination angle effects (high-level).
- Web Mercator vs UTM; implications for distance/area and visualization.

### Astronomy (analogy layer)

- Light curves: sampling cadence, gaps, detrending concept (high-level).
- Noise & SNR intuition: shot/read noise analogies; why smoothing helps.
- Photometric calibration & extinction/airmass as conceptual cousins of atmospheric correction.
- Basic change/variability detection (e.g., CUSUM/Lomb-Scargle-just the idea for now).