pebbs.app

Sunlight and Shade Visualisation App for Individual Properties

Executive summary

This concept—an app that visualises the sun's path and resulting sunlight/shade on (and around) a specific house across time (minute-by-minute time-lapse, date/season selection), embedded in a familiar map UI—has clear demand signals across both consumers (home buyers, renovators, gardeners) and professionals (estate agents, architects, solar installers). Competing products prove feasibility: consumer-grade "sun tracker" apps already provide sun path overlays and simple shadow estimates (often using AR + 2D map views), while dedicated shadow-mapping products simulate terrain/building/tree shadows on a map; professional AEC/solar tools run accurate "sun hours" / shading studies but are not streamlined for home-buying workflows. [1]

The decisive question is not "can we compute the sun's position?"—solar position algorithms are mature and extremely accurate. [2] The decisive questions for product quality and defensibility are: (a) what 3D environment model you can legally use, globally, at sufficient resolution and freshness (terrain + buildings + vegetation), (b) what accuracy you can honestly claim given data gaps (trees, building heights, renovations), and (c) whether you can deliver fast, intuitive, trustworthy UX without violating mapping/imagery licence terms (especially restrictions around caching, extraction, and Street View content). [3]

A practical MVP path exists that avoids the hardest parts (high-fidelity trees + interior daylight) while still delivering strong value in under ~8–12 weeks of focused work: **map-based address search → property outline + orientation → sun path + time slider → ground/roof shadow overlay from extruded building geometry → "sun-hours" summary metrics**. Higher fidelity (LiDAR/DSM-based trees + terrain ray casting, façade sun-hours, and interior daylight estimates) is best treated as a second phase, driven by data availability by region and clear, cautious "confidence" indicators. [4]

Target users and core product features

Target users and decision workflows

Home buyers and renters (including "garden-first" buyers) typically care about: direct sun in key areas (garden/patio, kitchen/living room windows), winter versus summer exposure (low sun angles in winter), and overshadowing from neighbouring buildings/trees—often before committing to viewings or renovations. Competing products explicitly mention "home seekers" and property decisions as use cases, which validates this positioning. [5]

Estate agents and property marketers care about: demonstrating "bright living spaces", supporting claims with defensible visuals, and reducing friction in remote viewings. Tools targeting "real estate" positioning already sell higher tiers for this segment (e.g., studio/project sharing, embedded scenes). [6]

A key UX implication: buyers need **fast answers and trustworthy summaries** ("Does this garden get sun after 3pm in October?"), while agents need **shareable, branded artefacts** (links, report PDFs, embeddable scenes) and auditability ("data sources, date/time assumptions, accuracy notes"). This split strongly supports a freemium consumer tier + paid professional tier. [7]

Core features requested and feasibility notes

Sun path overlay (map + AR/compass modes). This is table-stakes: multiple consumer apps show hourly sun direction intervals, solstice/equinox paths, and map views. [8]

Time-lapse shadow simulation (scrubbable time slider). Shadow-mapping tools describe real-time shadow calculations on a map and can summarise sun/shade hours over a day or year. [9]

Date selection and seasonal variation. Competitors explicitly support choosing arbitrary dates and showing yearly path envelopes; solar algorithms and NOAA calculators support arbitrary date/time at a location. [10]

Roof and ground shading. This is the differentiator for property use cases. High-quality results require building heights/roof geometry plus surrounding obstructions; professional solar tools highlight LiDAR-assisted modelling and shading reports as premium features, underscoring both value and implementation complexity. [11]

Interior light estimates (if feasible). Fully automated interior daylight prediction is not generally feasible without interior geometry, window layout, glazing, and materials; accurate daylight simulation typically uses physically-based tools (e.g., Radiance) and detailed scene descriptions. [12] A feasible product compromise is: (a) façade/window exposure proxies (sun-hours on façades, horizon obstruction angles), and (b) optional user-guided inputs (approx. window positions, room depth) leveraging established rules-of-thumb as a coarse indicator, clearly labelled as an estimate. [13]

Inputs required for credible results

At minimum (MVP-grade):
Location (lat/long), **time zone**, **timestamp/date**, and **a building footprint + approximate height and orientation**. The sun position component is straightforward; the environment model is the hard part. [14]

For higher fidelity (recommended "phase 2" inputs):
Elevation/terrain model (DEM/DTM), **surface model** (DSM including buildings/trees), **neighbouring building footprints and heights**, and **vegetation/obstacle geometry**. LiDAR point clouds and LiDAR-derived DEMs are highlighted by major national providers as sources for buildings, vegetation, and ground representation. [15]

For professional/regulated reporting (optional):
Cadastral parcel boundaries to anchor "this is the parcel" and support agent workflows; availability and licensing vary heavily by country (e.g., INSPIRE parcel datasets in parts of Europe). [16]

Solar position algorithms and shadow casting methods

Solar position algorithms and what "accuracy" really means

Solar position computations (solar azimuth and elevation/zenith as a function of time and location) are well-solved:

- **NOAA-style equations** provide practical, well-documented computations for equation of time, declination, and derived azimuth/elevation; NOAA states accuracy expectations for sunrise/sunset calculations and notes atmospheric variability impacts observed values. [17]
- **NREL Solar Position Algorithm (SPA)** is a high-precision reference approach; NREL documentation reports uncertainty of approximately ±0.0003° over a very wide year range (−2000 to 6000). [18]
- Many libraries implement these ideas (e.g., SunCalc is a BSD-licensed JavaScript library for sun position and phases); these are commonly sufficient for consumer UX and even many professional previews. [19]

However, for your app, **astronomical accuracy is almost never the limiting factor**. The dominant error sources are typically: (a) environment geometry quality (heights, trees), (b) coordinate/vertical datum mismatches, and (c) terrain/vegetation changes since capture. This is consistent with shadow-mapping vendors explicitly cautioning that maps are models/estimates, not "truth". [20]

Shadow casting approaches and trade-offs

A robust architecture usually supports **three modelling tiers**, chosen dynamically based on available data and device performance.

Tier A: Analytic shadow polygons from extruded buildings (fastest, MVP-friendly).
Mechanism: extrude the subject building footprint (and optionally nearby footprints) to heights; compute shadow polygons by projecting roof/edge vertices onto the ground plane along the sun vector; render as 2D overlay. This can be highly responsive and works well for "garden shade from neighbour's house" scenarios when heights are reasonable.

Limitations: ignores terrain variation and tree canopies unless separately modelled. Quality depends on building heights. Datasets that include height and roof attributes significantly improve this tier. [21]

Tier B: Raster DSM/DTM ray marching (balanced fidelity; global varies).
Mechanism: use a DSM (surface heights including buildings/vegetation) and optionally a DTM/DEM (ground) to perform line-of-sight checks per pixel or per sample point (e.g., every 1–2 metres), accumulating sun-hours over a day/year.

This is similar in spirit to "sun hours" analyses in AEC tooling, which explicitly uses ray tracing and grid sampling at regular intervals (e.g., checks every 6 minutes) to accumulate direct sun totals. [22]

Limitations: requires high-resolution DSM for convincing results; global 30m DSMs (e.g., Copernicus GLO-30) are often too coarse for a single property with small trees/buildings. [23]

Tier C: 3D engine shadows (visually compelling; licensing-sensitive).
Mechanism: render a 3D scene (terrain + buildings + trees) and enable shadow mapping in the renderer. For example, Cesium documentation notes globe shadow settings and performance implications:

enabling terrain casting shadows can re-render terrain from the light's perspective and may impact performance. [24]

Limitations: if the underlying 3D scene is sourced from restricted providers (e.g., photorealistic tiles), you may be constrained to "visualisation only", with prohibitions on caching and on deriving/extracting geodata from the tiles. [25]

Interior light estimates: what is feasible

Fully automated "interior daylight" (lux levels in rooms) generally requires: window geometry, glazing properties, room dimensions, interior surface reflectances, and external obstructions; accurate workflows often rely on physically-based ray tracing (Radiance is explicitly described as an accurate lighting simulation toolset). [26]

A realistic product approach is a **three-step ladder**:

1. **Façade/window exposure proxy**: compute direct sun-hours on building façades at sample points. This is aligned with professional tools that present sun-hours on façades/ground. [22]
2. **User-declared windows**: let the user mark window positions on the façade; infer sun penetration times (still approximate).
3. **Premium "room model" mode**: accept a simple room box (depth/height), approximate glazing transmittance, and compute coarse daylight factor bands; daylight factor is widely defined as indoor illuminance relative to outdoor, and is used in UK guidance with indicative thresholds (e.g., 2% and 5% bands) as a design heuristic. [27]

This should be framed carefully as *indicative*, not a substitute for a professional daylight assessment.

Data sources, mapping APIs, licensing limits, and compliance

Prioritised data sources and how to obtain/licence them

The following prioritisation is designed to maximise global coverage while preserving legal clarity and enabling an MVP without negotiating country-by-country data deals.

Priority	Data need	Candidate source	Coverage and key attributes	Licence / constraints	Practical acquisition path
Highest	Building footprints **with heights/ roof attributes**	Overture buildings dataset	Global "building" + "building_part" with attributes including height, floors, roof shape/ direction/ material in schema; downloadable from S3/Azure; Overture documents explicit access paths. [28]	ODbL share-alike (noted as driven by OpenStreetMap licence compatibility); requires careful compliance for derivative databases. [29]	Periodic bulk ingest + tiling into your own vector tile service; keep provenance metadata.
High	Footprints (permissive licence, extremely large coverage)	Microsoft GlobalMLBuildingFootprints	1.4B buildings detected from Bing imagery (2014–2024) and released for download; repo states licence as CDLA Permissive 2.0. [30]	CDLA-Permissive (more permissive than ODbL); still needs attribution/ notice retention per licence text (not reproduced here). [30]	Bulk download from GitHub; pre-tile (PMTiles/ MBTiles) for fast client queries.
High	Terrain/ elevation baseline	Copernicus DEM GLO-30 / GLO-90	Worldwide DSM at 30m/ 90m; official Copernicus docs indicate free licence; licence PDF defines terms. [31]	"Free licence" but must follow specific licence terms (redistribution/ attribution clauses per licence). [32]	Download from Copernicus Data Space / cloud mirrors; precompute terrain tiles. [33]

Priority	Data need	Candidate source	Coverage and key attributes	Licence / constraints	Practical acquisition path
Medium	Terrain/ elevation alternative	NASA SRTM / NASADEM	SRTM provides ~30m data; NASADEM is a reprocessing of SRTM with improved accuracy via auxiliary datasets. [34]	Licensing varies by distribution platform; confirm for your chosen host (e.g., NASA Earthdata vs third-party mirrors). [34]	Use NASA Earthdata cloud access or a vetted mirror; cache your own derived tiles. [35]
Region-dependent (premium fidelity)	LiDAR point clouds / LiDAR-derived DEM/DSM	USGS 3DEP (example of open national LiDAR)	USGS describes LiDAR point clouds representing buildings, vegetation, and ground; 3DEP products are free and "without use restrictions". [36]	Public domain / no restrictions for USGS 3DEP products (per USGS statements). [37]	Use USGS LidarExplorer / AWS Open Data for efficient streaming; generate DSM/DTM for shadow runs. [38]
Optional (workflow enhancement)	Parcels / cadastral polygons	INSPIRE cadastral datasets (example: UK Land Registry polygons under OGL)	HM Land Registry provides "INSPIRE Index Polygons" for England/ Wales; download portal indicates Open Government Licence v3.0. [39]	Often open (e.g., OGL in UK) but varies widely by jurisdiction; must track local terms. [40]	Integrate as optional layer per country; do not assume global availability.

A practical MVP can be built without LiDAR by using **Overture heights** (where present) + fallback heuristics. Overture's schema explicitly includes height, roof shape/direction, and floors, which is unusually valuable for a global dataset and directly supports the "extruded footprint" and basic roof modelling tiers. [41]

Mapping APIs and licensing: what you can and cannot do

Because your app's core value involves *derived* sunlight/shade information, you must be explicit about whether a given map/imagery provider permits:

1) caching/persistent storage,
2) computation/analysis beyond display, and
3) mixing with non-provider maps or overlays.

Key examples:

- **Google Maps Platform Map Tiles / Photorealistic 3D Tiles.** Google's Map Tiles API policies state you must not pre-fetch, index, store, or cache content except as allowed under limited conditions. [42] Policies also include restrictions on extracting/tracing/deriving 3D objects from photorealistic tiles, even while allowing overlays of your own 3D objects not derived from those tiles. [43] Google's own blog post on photorealistic 3D tiles points to explicit prohibitions including "image analysis" and "geodata extraction or resale" from imagery. [44]

- **EEA-specific terms can materially change behaviour.** Google's Solar API documentation warns that for EEA billing addresses effective 8 July 2025, certain Solar API content will no longer be returned. [45] EEA service-specific terms also include a "No Use With any Map" restriction for Street View Tiles (in the Map Tiles API). [46] This implies you must treat "Google-based street-level tiles everywhere" as *not globally uniform* and plan fallbacks.

- **Google Solar API (environment API) is powerful but purpose-constrained.** Solar API provides "buildingInsights" and "dataLayers" endpoints for rooftop solar potential evaluation. [47] Its service-specific terms include caching limits (30 days) for Solar Data and define permitted uses tied to energy system feasibility/design/transactions. [48] For your concept, Solar API is most defendable if positioned around "energy efficiency / solar readiness" rather than general home "niceness" claims, unless you obtain explicit permission/contractual coverage.

- **Mapbox offline and caching.** Mapbox documentation discusses offline map regions and constraints (e.g., tile pack limits and billing implications for offline use). [49] Mapbox's legal terms govern broader constraints; exact allowances depend on your contract and product. [50]

- **Cesium-based stacks for 3D + shadows.** CesiumJS is under Apache 2.0 and is free for commercial use; shadows and sun lighting are first-class features, with documentation explicitly discussing shadow behaviour and performance implications. [51] If you ingest your own open datasets into 3D Tiles, you can avoid imagery-provider restrictions, but you inherit the obligations of your data licences (e.g., ODbL). [52]

Privacy and legal issues

Your app touches two sensitive areas: (a) **precise location/property inference**, and (b) **street-level imagery**.

- Under UK GDPR guidance, personal data is information relating to an identified or identifiable natural person, potentially including location data. [53] You therefore need privacy-by-design: minimise stored precise addresses, use ephemeral processing where possible, and provide clear user-facing privacy policy and lawful basis. (This is especially important if you support agent accounts with client property lists.)

- For Street View / street-level imagery, Google's brand resource guidelines explicitly state you may not screenshot Street View imagery or remove it from embedded sources. [54] Google's Street View Static API policies also highlight that content caching/storage is generally prohibited with limited exceptions (e.g., panorama IDs). [55] If you include street-level context, ensure you comply with the relevant API's attribution, embedding, and caching rules.

A product positioning that emphasises "sunlight planning" (gardens, solar, energy efficiency) rather than "surveillance of private homes" also reduces misuse risk; you should still implement abuse controls (rate limiting, anomaly detection, and clear acceptable-use terms).

Competitive landscape

The market splits into four clusters: (1) **sun path trackers** (AR + map, little/no environment modelling), (2) **shadow-mapping map apps** (simulate shadows in 3D with varying data quality), (3) **solar/AEC professional tools** (accurate shading analyses, expensive, heavy workflows), and (4) **platform APIs** (solar/roof insights, licensing constraints).

Competitive landscape table

Note on URLs: to respect the "no raw URLs outside code" constraint, the "URL ref" column maps to a code block of URLs immediately after the table.

Name	Platform	Key features for this concept	Pros / cons (for your target users)	Pricing signal	URL ref
Shadowmap	Web + iOS	Global 3D sunlight/ shadow simulation; plans explicitly include "home seekers" and "real estate"; shareable projects on higher tiers. [56]	**Pros:** clearly aligned with "property/ daylight" use cases; pricing ladder already exists. **Cons:** differentiation may require better local accuracy, clearer confidence scoring, or unique workflow integrations for UK/EU agents. [57]	Free tier; paid tiers shown as numeric monthly/ annual amounts and "Studio" plan. [57]	U1

Name	Platform	Key features for this concept	Pros / cons (for your target users)	Pricing signal	URL ref
ShadeMap	Web	Simulates mountain/building/tree shadows for any date/time; computes sun/shadow hours and annual summaries; offers developer toolkit positioning. 58	**Pros:** “works online” framing; strong for “any place/time” UX. **Cons:** pricing not clearly visible in extracted content; fidelity/inputs and enterprise terms may be opaque for buyers/agents. 59	Free to use without login per third-party toolkit; API requires key. 60	U2
Sun Seeker	iOS / Apple Watch	Map + compass + AR sun path; explicitly mentions real estate buyers checking sun exposure before purchase. 61	**Pros:** cheap, simple mental model (“where will sun be”). **Cons:** does not model *actual* shadows from trees/buildings; limited for “will this patio be shaded”. 61	One-off app price shown (US store). 61	U3
Sun Surveyor	iOS / Android	AR projections of sun/moon paths; interactive map with Street View; includes “sun shadow information” (shadow length). 62	**Pros:** widely trusted “planner” tool; good for agents doing on-site demos. **Cons:** shadow modelling is mostly geometric/heuristic, not property-context simulation. 62	Paid app price shown (US store). 63	U4
Sun Tracker AR	iOS	Map + AR sun path; includes a “3D shadow simulator” for a user-placed object and can compute time a target spot is shaded by that object. 64	**Pros:** demonstrates “spot-level” shade UX. **Cons:** user-placed object ≠ full environment model; not a “house + neighbours” simulator. 64	Free with in-app purchases; premium unlock described. 64	U5

Name	Platform	Key features for this concept	Pros / cons (for your target users)	Pricing signal	URL ref
SunCalc (SunCalc.net + library)	Web + OSS library	Interactive sun path map and yearly path envelope; open-source JS library exists (BSD). [65]	**Pros:** proves standard UX patterns (sun curve + slider). **Cons:** not a property shading product; no building/tree occlusion by default. [65]	Free / open source. [19]	U6
Google Earth Pro (Sunlight mode)	Desktop	Desktop app is explicitly free; community references a "Sun"/ sunlight feature in Earth Pro workflows. [66]	**Pros:** familiar exploration + 3D context; good baseline for user expectations. **Cons:** not a specialised property sunlight decision tool; not packaged for estate agent sharing/ reporting. [66]	Free (explicit). [67]	U7
Aurora Solar	Web (B2B SaaS)	Remote site assessment, LiDAR-based shade analysis, AI-assisted 3D modelling; "bankable shade reports" called out in pricing/features. [68]	**Pros:** high accuracy expectations; strong precedent for monetising shading. **Cons:** solar-installer focus; likely too heavy/expensive for home buyers; not a pure "home daylight" product. [68]	Pricing page emphasises tiered plans; detailed public price varies by region/ contract. [69]	U8
HelioScope	Web (B2B SaaS)	C&I PV design; pricing page lists plans; Pro plan includes LiDAR-assisted modelling and AI obstruction detection. [70]	**Pros:** clear price points; strong "obstructions/ shading" emphasis. **Cons:** not aimed at home buying UX; PV workflows dominate. [70]	$159/mo basic; $259/ mo pro; enterprise custom (official page). [71]	U9

Name	Platform	Key features for this concept	Pros / cons (for your target users)	Pricing signal	URL ref
OpenSolar	Web + mobile	Free-of-charge positioning; support docs describe shading analysis relying on 3D roof geometry and simulated shadows; shading reports include annual/monthly/hourly sun access concepts. [72]	**Pros:** demonstrates a “free” model subsidised by partnerships; shading concepts + reporting patterns. **Cons:** solar-pro workflow; not a consumer daylight product. [73]	“100% free of charge” messaging. [74]	U10
Google Maps Platform Solar API	API	Building insights + data layers for rooftop solar potential; usage/billing is pay-as-you-go; coverage stated as hundreds of millions of buildings in dozens of countries. [75]	**Pros:** high-quality rooftop/shading-related datasets for solar. **Cons:** licensing/ permitted-use constraints; EEA limitations and caching rules; may not fit “home buying daylight” without careful positioning/ permission. [76]	Pay-as-you-go with free caps; SKU pricing published. [77]	U11

```
U1  https://shadowmap.org/pricing
U2  https://shademap.app/
U3  https://apps.apple.com/us/app/sun-seeker-sunlight-tracker/id330247123
U4  https://apps.apple.com/us/app/sun-surveyor-sun-moon/id525176875
U5  https://apps.apple.com/us/app/sun-tracker-ar/id1460472433
U6  https://github.com/mourner/suncalc  (library)  |  https://
suncalc.net/ (web tool)
U7  https://www.google.com/intl/en/earth/versions/
U8  https://aurorasolar.com/pricing/
U9  https://helioscope.aurorasolar.com/pricing/
U10 https://www.opensolar.com/
U11 https://developers.google.com/maps/documentation/solar/overview
```

Competitive gap you can exploit: Most consumer apps answer “where is the sun?” but not “where will shadows fall on *this exact property*?”. Shadow-mapping apps answer that but may lack (a) region-specific trust signals, (b) estate-agent workflows (CRM, shareable branded reports, compliance), and (c) transparent confidence scoring and user-correctable inputs. Professional solar/AEC tools have strong modelling but are over-scoped and not optimised for home-buying. [78]

Technical architecture, recommended stack, and implementation plan

Reference architecture

```
flowchart LR
  U[User: buyer/agent] --> FE[Web/PWA or Mobile App]
  FE --> MAP[2D/3D Map Renderer]
  FE --> TIME[Time slider + date/season UI]
  FE --> AUTH[Auth/Billing]

  FE --> API[Backend API]
  API --> GEO[Geocoder/Search]
  API --> PROP[Property Resolver: footprint/parcel]
  API --> SUN[Sun Position Service]
  API --> SHADOW[Shadow Engine]
  API --> CACHE[Result Cache + CDN]

  PROP --> BLDG[Buildings Dataset]
  PROP --> PARCEL[Parcel/Cadastre (optional)]
  SHADOW --> TERR[Terrain/DSM/DTM Tiles]
  SHADOW --> OBST[Obstructions: nearby buildings/trees]

  BLDG -->|bulk ingest| ETL[ETL + Tiling Pipeline]
  TERR -->|bulk ingest| ETL
  ETL --> TILE[Vector/Raster Tile Store]

  MAP -->|tiles| TILE
  SHADOW -->|reads| TILE
  SHADOW -->|outputs: polygons/rasters| CACHE
  FE -->|renders overlay| MAP
```

This architecture supports multiple computation modes:

- **Client-side sun position + simple shadows** (Tier A), with optional server-side “sun-hours” accumulation to reduce device load.
- **Server-side raster ray marching / GPU ray tracing** (Tier B/C) where high-res DSM is available and you want consistent results across devices, similar to AEC tool patterns that accumulate sun exposure by repeated ray checks. [79]

Recommended tech stack

Frontend (MVP):
- Web-first (React + TypeScript) as a PWA to reach desktop and mobile quickly; map UI patterns are easier to iterate than full native builds. - 2D map: MapLibre GL JS (open ecosystem) or Leaflet; 3D add-on path via CesiumJS if required. (MapLibre is commonly used as a permissive alternative in the Mapbox-GL ecosystem after licensing changes, though your tile/data provider terms still apply.) [80]

Backend:
- Node.js or Python (FastAPI) for API orchestration; heavy geospatial work in Python (GDAL/rasterio) or in a dedicated service (Rust/C++ for performance). - PostGIS for spatial indexing and querying footprints/nearby obstructions; object storage (S3-compatible) for tile archives and precomputed rasters.

Shadow engine:
- Tier A: CPU geometry ops (polygon projection + union, e.g., GEOS).
- Tier B/C: GPU acceleration for raster ray casting when creating heatmaps or annual sun-hours; align with ray-tracing approaches used in professional products (noting that your implementation and data quality still govern accuracy). [81]

Data pipeline:
- Bulk ingest Overture/Microsoft/Copernicus into tiled formats; Overture explicitly publishes S3/Azure locations and schemas. [82]

Implementation plan with milestones and cost bands

The effort below assumes a small, focused team and avoids "infrastructure rabbit holes" by using managed services where it materially reduces delivery risk. Cost bands are indicative and vary massively by region/rates; the point is scope control.

MVP scope (aim: 6–10 weeks to usable beta)
Milestones: 1. **Sun path MVP**: address/lat-long selection, time slider, date picker, solstice/equinox presets, sun azimuth/elevation overlay. Use NOAA equations or a proven library; validate against NOAA calculator outputs. [83]
2. **Property geometry**: fetch building footprint (+ height where available) using a global dataset like Overture; fallback to a permissive footprint set (Microsoft) with heuristics for height when absent. [84]
3. **Shadow overlay Tier A**: extruded-footprint shadow polygons on ground; "sun-hours on selected area" by sampling times across a day (e.g., 10-minute steps).
4. **Shareability**: link sharing with embedded timestamp/date; export a simple "Sunlight summary" report (PDF or share page).
5. **Trust UX**: show "data confidence" (e.g., height known vs estimated; last-updated for buildings dataset) and a manual correction UI for height/roof direction.

Estimated effort (very rough): 1 full-stack engineer + 1 geospatial/graphics engineer for ~6–10 weeks, plus design time.
Rough cost band (build only): **£30k–£120k** depending on whether you use contractors versus in-house and how much geodata ETL is done upfront.

Expanded product (aim: 3–6 months)
Add: - High-res DSM/DTM ingestion where available (starting with a few high-value countries/metros) and Tier B/C sun-hours heatmaps. - Vegetation/trees modelling (LiDAR-derived or curated sources), façade sun-hours, and "seasonal comfort" metrics. - Estate agent tools: branded property pages, portfolio comparison dashboards, audit logs. - Optional Solar API integration if you position around solar/energy outcomes and can comply with permitted-use and caching constraints; monitor EEA availability changes. [85]

Estimated effort: 3–6 engineers (incl. geospatial + graphics) over 1–2 quarters.
Rough cost band: **£200k–£800k** for a serious multi-region product + data pipeline + compliance work, excluding major commercial data purchases.

Ongoing costs (order-of-magnitude): - Mapping + geocoding: subscription/pay-as-you-go can be meaningful; Google publishes subscription tiers and per-SKU pricing; Mapbox pricing is usage-based with free tiers. [86]
- Storage/compute: grows with raster/DSM + caching and report generation.
- Data licensing/compliance: potentially the largest long-term risk if you rely on restricted imagery for anything beyond display. [87]

Sample API calls and algorithm pseudocode

Example: Google Solar API "buildingInsights" request (if used for solar/roof contexts)

(You must comply with Solar API policies, permitted-use terms, and billing; quotas and SKU pricing are published. [88])

```
GET https://solar.googleapis.com/v1/buildingInsights:findClosest
  ?location.latitude=51.5074
  &location.longitude=-0.1278
  &requiredQuality=HIGH
  &key=YOUR_API_KEY
```

Example: OSM Overpass footprint query (for fallback / validation)

(Overpass public infrastructure has shared capacity; usage guidance notes scale constraints and encourages running your own instance for heavy demand. [89])

```
POST https://overpass-api.de/api/interpreter
Content-Type: application/x-www-form-urlencoded

data=[out:json];
way["building"](around:50,51.5074,-0.1278);
out geom;
```

Sun position pseudocode (NOAA-style)

NOAA publishes "general solar position calculations" including fractional year, equation of time, and declination approximations. [90]

```
function solar_position_NOAA(datetime_utc, lat_deg, lon_deg):
    # 1) Convert datetime to day_of_year and fractional hour (UTC)
    N = day_of_year(datetime_utc)
    hour = datetime_utc.hour + datetime_utc.minute/60 + datetime_utc.second/
3600

    # 2) Fractional year (gamma) in radians
```

```
    gamma = 2*pi/365 * (N - 1 + (hour - 12)/24)

    # 3) Equation of time (minutes)
    eqtime = 229.18 * (0.000075
                      +0.001868*cos(gamma)
                      -0.032077*sin(gamma)
                      -0.014615*cos(2*gamma)
                      -0.040849*sin(2*gamma))

    # 4) Solar declination (radians)
    decl = (0.006918
           -0.399912*cos(gamma)
           +0.070257*sin(gamma)
           -0.006758*cos(2*gamma)
           +0.000907*sin(2*gamma)
           -0.002697*cos(3*gamma)
           +0.00148*sin(3*gamma))

    # 5) Time offset and true solar time
    time_offset_min = eqtime + 4*lon_deg   # ignoring DST, use UTC
    tst_min = hour*60 + time_offset_min

    # 6) Hour angle (deg)
    ha_deg = (tst_min/4) - 180

    # 7) Solar zenith
    lat = radians(lat_deg)
    ha  = radians(ha_deg)
    cos_zen = sin(lat)*sin(decl) + cos(lat)*cos(decl)*cos(ha)
    zen = arccos(cos_zen)
    elev = (pi/2) - zen

    # 8) Solar azimuth (one common convention; ensure you standardise)
    az = atan2( sin(ha),
                cos(ha)*sin(lat) - tan(decl)*cos(lat) )
    az_deg = (degrees(az) + 180) mod 360

    return az_deg, degrees(elev)
```

For highest precision (and to reduce edge-case risk at extreme latitudes), call an SPA implementation; NREL documents SPA uncertainty and validity range. [91]

Shadow casting pseudocode (Tier A: extruded footprint)

```
function shadow_polygon_extruded(building_polygon_ENU, height_m,
sun_dir_ENU_unit):
    # sun_dir points from origin towards the sun; for casting shadows,
project along -sun_dir
    if sun_dir_ENU_unit.z <= 0:
        return EMPTY  # sun below horizon -> no direct sun
```

```
    shadow_points = []
    for vertex in building_polygon_ENU.vertices:
        # Assume roof vertices at z = height_m (flat roof approximation)
        roof_v = (vertex.x, vertex.y, height_m)

        # Find intersection with ground plane z=0 along -sun_dir
        t = height_m / sun_dir_ENU_unit.z
        ground_p = roof_v - t * sun_dir_ENU_unit

        shadow_points.append(ground_p.xy)

    # The simple approach returns a projected polygon; robust approach unions
roof edges + walls.
    return polygon_union_and_clean(shadow_points)
```

Tier B raster casting (DSM ray marching) uses repeated line-of-sight checks from sample points to the sun direction; professional tools describe similar repeated checks to accumulate sun-hours. [92]

Risks and mitigations

Data accuracy and user trust risk

Risk: Missing/incorrect building heights, outdated vegetation, renovations, and seasonal tree canopy changes can dominate error, making "exact sunlight" claims risky. [93]
Mitigation: Provide a visible confidence score per result ("height known" vs "estimated", "trees modelled" vs "not modelled"), allow user correction of height/roof direction, and present outcomes as estimates with clear assumptions.

Licensing and platform dependency risk

Risk: Restrictions on caching, analysis, and geodata extraction can break core features if you depend on restricted map tiles/3D photorealistic content; EEA-specific terms can remove content or change allowed uses. [94]
Mitigation: Build on open datasets for geometry/shading, treat commercial tiles as *visual basemaps only*, and design a "provider abstraction layer" so you can switch basemaps/tiles without rewriting the shadow engine.

Privacy, misuse, and regulatory exposure

Risk: Processing addresses and showing "when a home is shaded/sunny" can be misused for inference; location data can be personal data under UK GDPR. [95]
Mitigation: Store minimal personal data; offer anonymous mode; encrypt sensitive user projects; rate limit and monitor scraping; publish clear ToS/Privacy Policy; avoid features that encourage targeting individuals; follow imagery rules (e.g., Street View no-screenshot guidance). [96]

Performance and cost blow-ups

Risk: Real-time shadowing over high-res DSMs and time-lapse/annual sun-hours can be compute-heavy; 3D shadows can be expensive in render pipelines and noted to impact performance. [97]

Mitigation: Progressive refinement (fast Tier A first, then "compute detailed sun-hours" async), LOD tiling, server-side caching, and strict quota controls. If using third-party APIs, enforce budgets/quotas (Google explicitly recommends quota limits to manage Solar API costs). [98]

Monetisation and go-to-market risk

Risk: Consumers may not pay much for "sun path" alone (because cheap apps exist), while pros demand defensible accuracy and sharing/reporting. [99]
Mitigation: Differentiate with (a) property-specific shade overlays and sun-hours summaries, (b) estate-agent collaboration features, (c) region-specific high-fidelity data add-ons, and (d) pricing aligned to stakeholder value (per-listing/per-report for agents; freemium for buyers). Shadowmap's tiering shows this pattern is already market-validated. [100]

---

[1] [62] [63] https://apps.apple.com/us/app/sun-surveyor-sun-moon/id525176875
https://apps.apple.com/us/app/sun-surveyor-sun-moon/id525176875

[2] [14] [17] [83] [90] https://gml.noaa.gov/grad/solcalc/solareqns.PDF
https://gml.noaa.gov/grad/solcalc/solareqns.PDF

[3] [25] [42] [43] [87] [94] https://developers.google.com/maps/documentation/tile/policies
https://developers.google.com/maps/documentation/tile/policies

[4] [15] [36] [38] https://www.usgs.gov/faqs/what-lidar-data-and-where-can-i-download-it
https://www.usgs.gov/faqs/what-lidar-data-and-where-can-i-download-it

[5] [6] [7] [10] [57] [100] https://shadowmap.org/pricing
https://shadowmap.org/pricing

[8] [61] [78] [99] https://apps.apple.com/us/app/sun-seeker-sunlight-tracker/id330247123
https://apps.apple.com/us/app/sun-seeker-sunlight-tracker/id330247123

[9] [58] https://shademap.app/help/
https://shademap.app/help/

[11] [69] https://aurorasolar.com/pricing/
https://aurorasolar.com/pricing/

[12] [26] https://www.energy.gov/cmei/buildings/articles/radiance
https://www.energy.gov/cmei/buildings/articles/radiance

[13] [22] [79] [81] [92] https://blogs.autodesk.com/forma/2025/12/19/how-to-analyze-your-first-proposal/
https://blogs.autodesk.com/forma/2025/12/19/how-to-analyze-your-first-proposal/

[16] https://inspire.ec.europa.eu/theme/cp
https://inspire.ec.europa.eu/theme/cp

[18] [91] https://docs.nrel.gov/docs/fy08osti/34302.pdf
https://docs.nrel.gov/docs/fy08osti/34302.pdf

[19] https://github.com/mourner/suncalc
https://github.com/mourner/suncalc

[20] [93] https://tedpiotrowski.svbtle.com/sun-and-shadow-maps-models-vs-reality
https://tedpiotrowski.svbtle.com/sun-and-shadow-maps-models-vs-reality

[21] [28] [29] [41] [52] [82] [84] https://docs.overturemaps.org/guides/buildings/
https://docs.overturemaps.org/guides/buildings/

23 31 https://dataspace.copernicus.eu/explore-data/data-collections/copernicus-contributing-missions/collections-description/COP-DEM
https://dataspace.copernicus.eu/explore-data/data-collections/copernicus-contributing-missions/collections-description/COP-DEM

24 97 https://cesium.com/downloads/cesiumjs/releases/1.74/Build/Documentation/Globe.html
https://cesium.com/downloads/cesiumjs/releases/1.74/Build/Documentation/Globe.html

27 https://en.wikipedia.org/wiki/Daylight_factor
https://en.wikipedia.org/wiki/Daylight_factor

30 https://github.com/microsoft/globalmlbuildingfootprints
https://github.com/microsoft/globalmlbuildingfootprints

32 https://docs.sentinel-hub.com/api/latest/static/files/data/dem/resources/license/License-COPDEM-30.pdf
https://docs.sentinel-hub.com/api/latest/static/files/data/dem/resources/license/License-COPDEM-30.pdf

33 https://egnos.gsc-europa.eu/sites/default/files/2025-07/Guide%20on%20Copernicus%20DEM%20GLO%20-%2030%20for%20Flight%20Procedure%20Design.pdf
https://egnos.gsc-europa.eu/sites/default/files/2025-07/Guide%20on%20Copernicus%20DEM%20GLO%20-%2030%20for%20Flight%20Procedure%20Design.pdf

34 35 https://www.earthdata.nasa.gov/data/catalog/lpcloud-srtmgl1-003
https://www.earthdata.nasa.gov/data/catalog/lpcloud-srtmgl1-003

37 https://www.usgs.gov/3d-elevation-program
https://www.usgs.gov/3d-elevation-program

39 https://www.gov.uk/guidance/inspire-index-polygons-spatial-data
https://www.gov.uk/guidance/inspire-index-polygons-spatial-data

40 https://www.nationalarchives.gov.uk/doc/open-government-licence/version/3/
https://www.nationalarchives.gov.uk/doc/open-government-licence/version/3/

44 https://mapsplatform.google.com/resources/blog/commonly-asked-questions-about-our-recently-launched-photorealistic-3d-tiles/
https://mapsplatform.google.com/resources/blog/commonly-asked-questions-about-our-recently-launched-photorealistic-3d-tiles/

45 47 75 76 85 https://developers.google.com/maps/documentation/solar/overview
https://developers.google.com/maps/documentation/solar/overview

46 https://cloud.google.com/terms/maps-platform/eea/maps-service-terms
https://cloud.google.com/terms/maps-platform/eea/maps-service-terms

48 https://cloud.google.com/maps-platform/terms/maps-service-terms/index-20240515
https://cloud.google.com/maps-platform/terms/maps-service-terms/index-20240515

49 https://docs.mapbox.com/help/dive-deeper/mobile-offline/
https://docs.mapbox.com/help/dive-deeper/mobile-offline/

50 https://www.mapbox.com/legal/tos
https://www.mapbox.com/legal/tos

51 https://github.com/CesiumGS/cesium
https://github.com/CesiumGS/cesium

[53] [95] https://ico.org.uk/for-organisations/uk-gdpr-guidance-and-resources/personal-information-what-is-it/what-is-personal-data/what-is-personal-data/
https://ico.org.uk/for-organisations/uk-gdpr-guidance-and-resources/personal-information-what-is-it/what-is-personal-data/what-is-personal-data/

[54] https://about.google/brand-resource-center/products-and-services/geo-guidelines
https://about.google/brand-resource-center/products-and-services/geo-guidelines

[55] [96] https://developers.google.com/maps/documentation/streetview/policies
https://developers.google.com/maps/documentation/streetview/policies

[56] https://shadowmap.org/
https://shadowmap.org/

[59] https://shademap.app/about/
https://shademap.app/about/

[60] https://bellingcat.gitbook.io/toolkit/more/all-tools/shademap
https://bellingcat.gitbook.io/toolkit/more/all-tools/shademap

[64] https://apps.apple.com/us/app/sun-tracker-ar/id1460472433
https://apps.apple.com/us/app/sun-tracker-ar/id1460472433

[65] https://suncalc.net/
https://suncalc.net/

[66] [67] https://www.google.com/intl/en/earth/versions/
https://www.google.com/intl/en/earth/versions/

[68] https://aurorasolar.com/design-mode/
https://aurorasolar.com/design-mode/

[70] [71] https://helioscope.aurorasolar.com/pricing/
https://helioscope.aurorasolar.com/pricing/

[72] [73] [74] https://www.opensolar.com/
https://www.opensolar.com/

[77] [88] [98] https://developers.google.com/maps/documentation/solar/usage-and-billing
https://developers.google.com/maps/documentation/solar/usage-and-billing

[80] https://www.maptiler.com/news/2021/01/maplibre-mapbox-gl-open-source-fork/
https://www.maptiler.com/news/2021/01/maplibre-mapbox-gl-open-source-fork/

[86] https://mapsplatform.google.com/pricing/
https://mapsplatform.google.com/pricing/

[89] https://dev.overpass-api.de/overpass-doc/en/preface/commons.html
https://dev.overpass-api.de/overpass-doc/en/preface/commons.html