

Project idea (name + one-line)

Greenfield Radar — Micro-Market Opportunity Detector

Find *exact places* in a city that are “white space” for a new shop by combining catchment isochrones, competitor density heatmaps, POI footfall proxies and a quick economic feasibility slider — all in one interactive map.

Why it's special: it fuses several high-value features from your doc into a single visual insight: instead of only showing POIs or isochrones, it *automatically grades small 100–500 m micro-tiles* and surfaces the best tiny pockets where demand > supply. That “micro-tile scoring + map-first UI + one-click demo” is visually striking and hackathon-friendly.

MVP (what you must build in 24 hours)

Core features (must-have for demo):

1. Map UI with ability to drop a candidate pin (or choose city sample).
2. Compute a catchment area for that pin (isochrone or simple drive/walk radius).
3. Nearby POIs & competitor fetch inside catchment (using Places API or a mock dataset).
4. Competitor density heatmap around the area.
5. Micro-tile scoring: divide catchment into a grid (e.g., 50m cells), compute an **Opportunity Score** per cell using:
 - competitor_count (penalty),
 - amenity_count and transit_count (positive),
 - estimated population density (if available) or POI proxies (positive),
 - commute accessibility factor (positive).
6. Show top 3 micro-tiles and an overall **Location Score** (0–100) with reasons (e.g., “Low competition, high transit, high amenity count”).

7. Demo mode with **sample city dataset** (so no paid API keys needed during hack).

Stretch goals (if time): single-vehicle route planner demo for deliveries from chosen site; “what if” slider to prioritize rent vs footfall; export PDF one-pager.

Why it's hackathon-wow

- Visual: dynamic heatmap + top micro-tiles is instantly demoable and looks polished.
 - Novelty: many demos show isochrones or competitor dots — few automatically score *micro-tiles* inside an isochrone and propose exact spots.
 - Business value: instantly answers “where exactly to open a kiosk/pop-up” — judges like clear impact.
-

Tech stack (fast, familiar, hackable)

- Frontend: React + Mapbox GL JS or Leaflet. Mapbox gives nice tiles; Leaflet + OpenStreetMap is free.
 - Backend: Node.js + Express (simple endpoints to call Places/Geocoding or serve mock data).
 - Data: Google Places / Mapbox Places (real), or CSV mock (fast). Use SQLite or in-memory JSON for demo data.
 - Deployment: run locally or a single Heroku / Vercel endpoint for backend.
 - Extras: small CSS framework (Tailwind) for quick UI.
-

Architecture (very small)

1. Frontend map → user places pin → POST with lat/lng.
 2. Backend: geocode (optional) → compute isochrone (or radius sampling) → call Places API for POIs and competitor names → compute tile grid + scores → return map layers + top tiles.
 3. Frontend renders heatmap + tile highlights + summary card.
-

Scoring formula (simple, transparent, easy to code)

(Provide as pseudocode — you can tune weights during hack)

Use min/max normalization per catchment. Keep weights adjustable with UI sliders for demo.

Note: normalization functions are simple $(\text{value} - \min)/(\max - \min)$. Keep numbers integers while demoing so judges can understand quickly.

24-hour build plan (hour by hour — follow this strictly)

No clarifying Qs — just build.

Hour 0 — 1: Project skeleton

- Create React app + Node/Express backend.
- Add Mapbox or Leaflet map to frontend.
- Wire simple backend endpoint

Hour 1 — 3: Map input, sample data & mock mode

- Implement “Drop pin / Use sample city” UI.
- Create sample dataset: competitor points, POIs in JSON for a 2×2 km area (you can extract from your research or fabricate realistic points).
- Implement radius fallback (e.g., 400m walk / 1000m drive).

Hour 3 — 6: Nearby POI + competitor retrieval & grid generator

- Backend endpoint that reads mock POIs within radius and returns them.
- Implement grid generator: split bounding box into $N \times N$ tiles (e.g., 20×20).
- Compute per-tile counts (POIs, competitors, transit stops).

Hour 6 — 9: Scoring & heatmap data

- Implement simple scoring function and normalization.
- Return geojson for heatmap + top 3 tiles.

Hour 9 — 12: Frontend visualization & UI polish

- Render heatmap (Mapbox/Leaflet heatmap layer) and tile polygons for top tiles.
- Add summary card: Location Score, top reasons, three recommended micro-tiles with quick stats.

Hour 12 — 15: Real API integration (optional)

- If you have time and API keys: swap mock POI fetch with Google Places / Mapbox Places calls.
- Otherwise keep mock dataset but add a “Connect API” mode for judges later.

Hour 15 — 18: Demo scenarios & pitch material

- Prepare 2 preloaded scenarios (e.g., busy mall street vs residential micro-pocket) and a 60-second pitch.
- Add slider to change weight priorities (competition vs footfall) to show the algorithm adapts live.

Hour 18 — 21: Polish & deploy

- Fix UI edge cases, add tooltips explaining score components.
- If possible, deploy backend to a free tier and host frontend on Vercel.

Hour 21 — 24: Final testing & presentation prep

- Run through demo script 3 times, record one short screencast (optional).
- Create one PDF export that summarizes the selected top location (for judges).

Demo script (90 seconds)

1. “Drop a pin here” — show catchment + POI dots.
2. Click “Score location” — heatmap burns up; top tile highlighted.
3. Show summary card: Location Score 82/100 — reasons (low competition, high transit).
4. Change slider to prioritize rent (competitor penalty ↓) — watch top tile change.

5. Click “Export one-pager” — download policy-ready PDF.

(Short, visual, repeatable — judges love manipulation + immediate outcome.)

Mock data & datasets to use

- Your uploaded doc suggests POI, isochrone and competitor ideas — use those as the feature set.
Location-Based Analytics Featur...
 - For population estimates use coarse census block numbers (or use POI count as a proxy for footfall).
 - Keep a prebuilt JSON for the hack to avoid rate limits.
-

What to show to judges for impact (must-have)

1. Live map with **heatmap + top tiles**.
 2. Clear numeric Location Score + 3 reasons (explainable).
 3. “Before/After” scenario showing competitor density causing a different choice.
 4. A one-page PDF export for a mock investor/owner.
-

Stretch ideas (nice to mention)

- Add phone-camera mode: scan a street block and auto-suggest best micro-tile for pop-up.
 - Live crowdsourced update: allow users to upvote observed footfall and tweak scores.
 - Fleet integration: pick optimal delivery hub inside selected tile and show route optimization (single-vehicle TSP).
-

Quick 60-second pitch (copy & paste in your presentation)

“Greenfield Radar finds the exact 50–200 m pockets in a neighborhood where opening a shop (or pop-up) is most likely to succeed. It combines catchment analysis, competitor density, amenity and transit presence and returns a simple, explainable *Location Score* and three exact micro-tiles to open in. Judges: instead of a vague ‘this area is good’, Greenfield Radar says *this block, this corner, these three meters* — demo in 60 seconds.”