Initial Requirement:

Let's start with what the project is.

* Client: Lake Pulse
* Intent: to capture data about every lake in the US, to easily search and surface data about each lake, and to create communities for each lake
* Project: a web portal with a primary map feature used for search, plotting and polygons, a community board, shopping features, AI chatbot, data/dashboard, and more (not all to start, but over time these features will be added).

We are not responsible for the lake data collection part, a separate partner is already doing that. We only need to build the portal UI.

We've already completed the first phase, which is a clickable demo of the portal. You can find it here:

<https://www.figma.com/proto/9JCFjrhjqvjwvXmKYtVcL8/Lake-Pulse?node-id=168-10&node-type=frame&m=dev&scaling=contain&content-scaling=fixed&page-id=0%3A1&starting-point-node-id=168%3A10>

The basic script you can use to walk through this demo is as follows:

* Start on Login page, click LOGIN button
* Search - click 'Enter a lake name' in search bar
* Search - view results, click 'Go to lake' under Elkhart
* Elkhart - view results, click 'Add this to my lakes' in upper right
* My Lakes - click Elkhart 'Go to lake' again
* Elkhart - see additional buttons in upper right. Then click My Lakes in top nav.
* My Lakes - click 'Go to lake' under Old Elk Lake
* Old Elk Lake - note new info and Layers feature, including button in upper right. Click Search
* Search - click Search by Map in sub nav. Note possibly if draggable map that updates search results. Click Toolbox.
* Toolbox - buy stuff
* Boathouse - AI conversational search

Related to the MVP, we do NOT need to worry about the Toolbox or Boathouse. We'll need to focus on server configuration, basic portal setup, user authentication/authorization, map search and simple results.

Assume that all of this will be spun up and built in AWS, so an understanding of EC2, RDS, IAM, Cognito, Docker and more will be essential here.

Technologies

* Auth = Amazon Cognito
* Page Analytics = Google Analytics
* Map = LeafletJS (<https://leafletjs.com/>) - this supports long/lat, point grouping and custom polygons
* Charting = Nivo, D3JS

Based on the current set of mocks, we'll need the following pages and features:

* Working Login page. We should work to capture user info as they navigate the site, integrate Google Analytics for this. In a future phase we can tie the anonymous data collected by GA to user data we may hold elsewhere (CRM).
* Search
  + Search by Name - select State, see all Lakes for that state, continue searching by lake name.
  + Search by Map - plot lat/long of lakes, drag to see individual or groups based on zoom, show included results in separate pane.
  + In either search, click link to go to Lake Page (paywall decisions potentially need to be made here)
* Individual Lake Page
  + Show lake on map (but likely no icons for sensors yet)
  + Show basic info
  + Show latest stats (decide how to score)
  + Show trends
  + Show weather (requires paid API)
  + NOT part of this phase - polygons, uploading data, adding data
* My Lakes
  + Collection of saved lakes with basic data
* Field Notes (subset of Lakes Page)
  + Unclear if required in Phase 3. I would recommend this gets pushed to future phase
  + If so, need a message board functionality tied to user logins
* Results (subset of Lakes Page)
  + Useful to prove breadth and depth of data
  + Keep simple to start, basic tabular data, possibly some charts
* Toolbox - not in phase
* Boathouse - not in phase