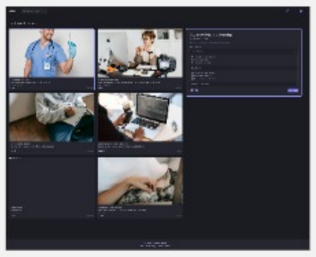
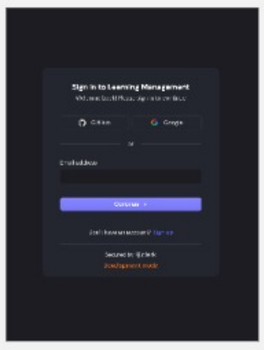
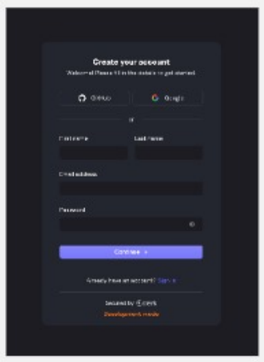
NOTE: these are screenshots of every page of the entire application, not figma because the app ended up very different from designs, hence why they don't look as good as they should for designs

Search Pages

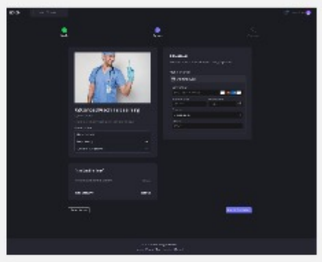


Auth Pages

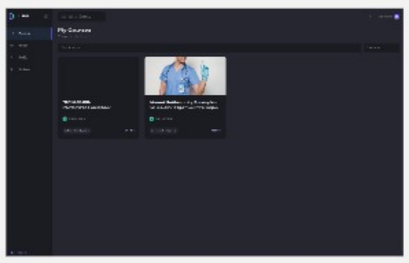


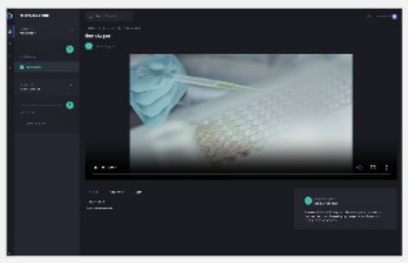


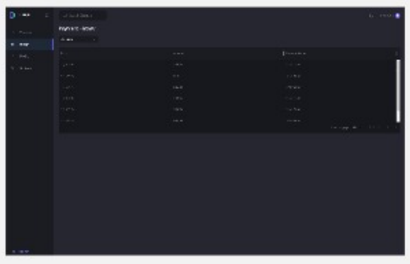
Checkout Pages

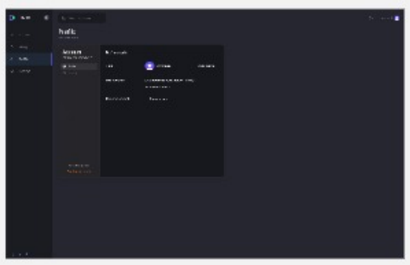


User Pages

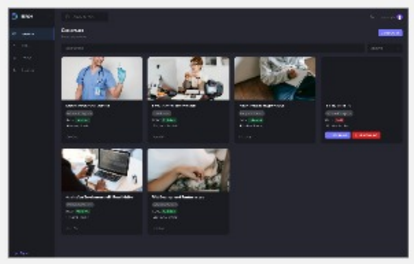


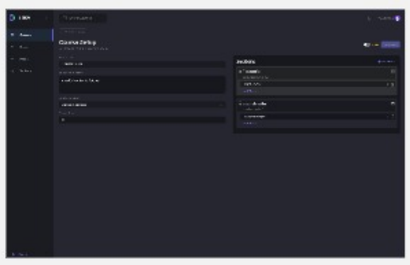


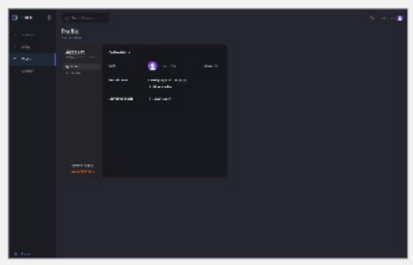


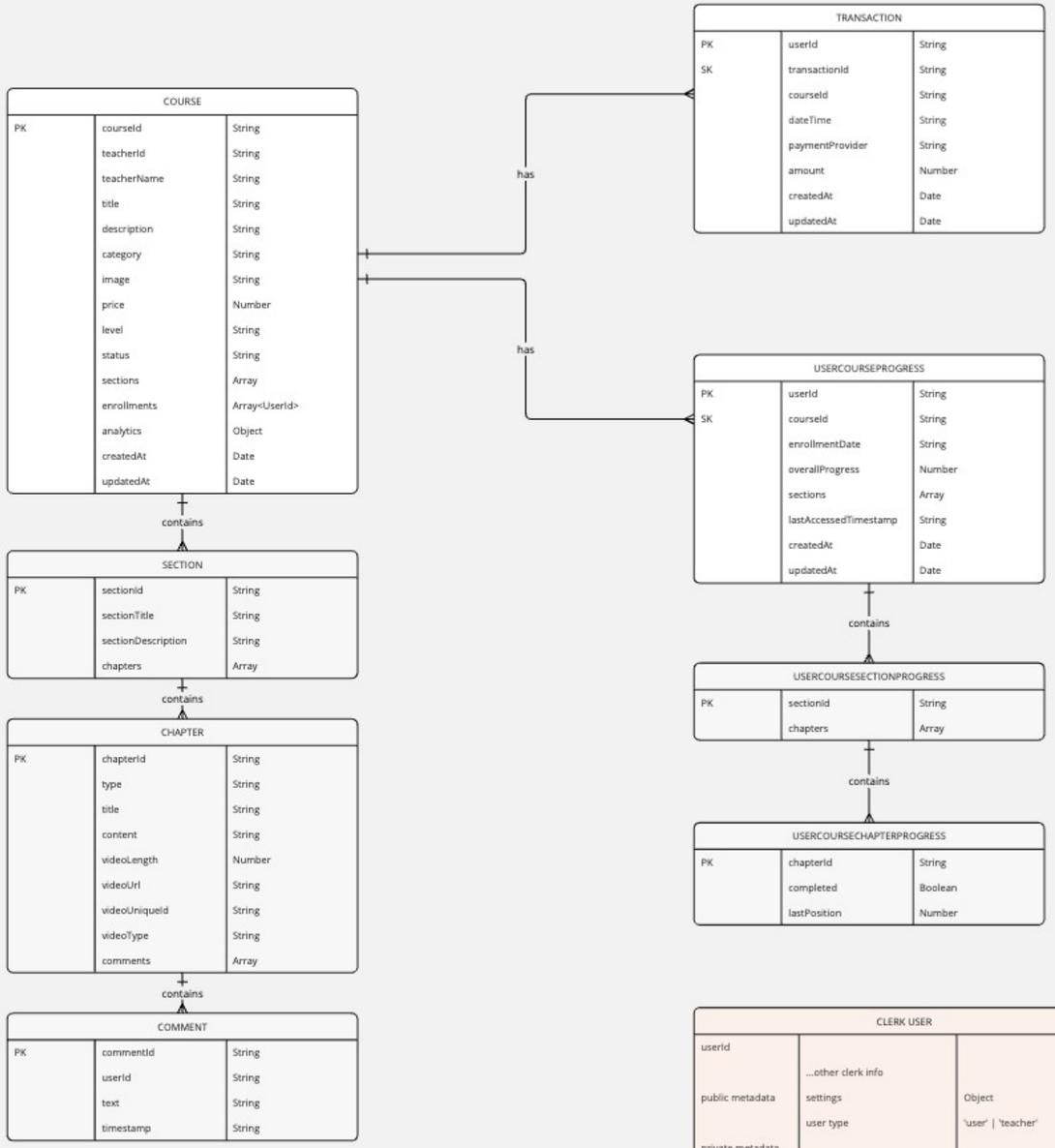


Teacher Pages









Why Choose DynamoDB?

The first choice in choosing a database is generally deciding between SQL vs NoSQL

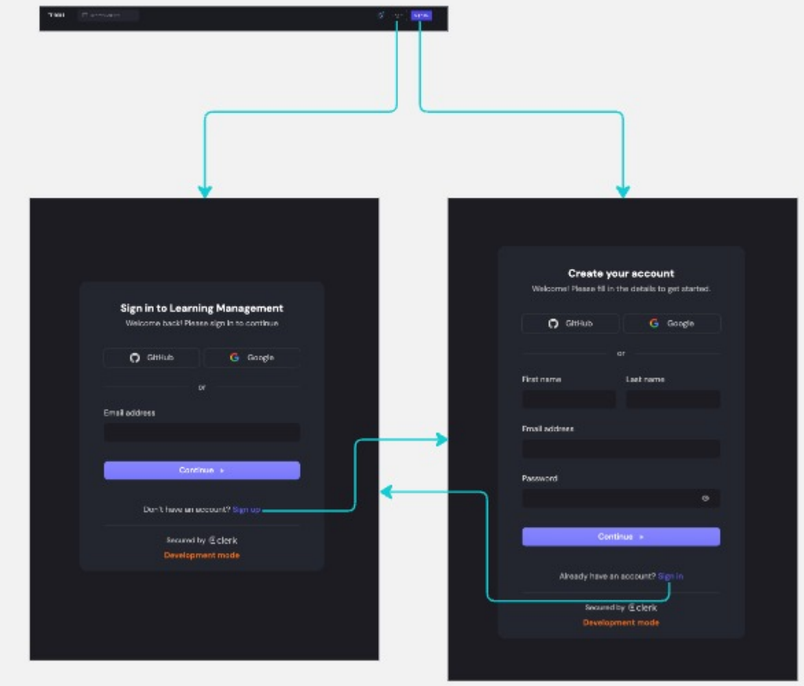
NoSQL vs SQL

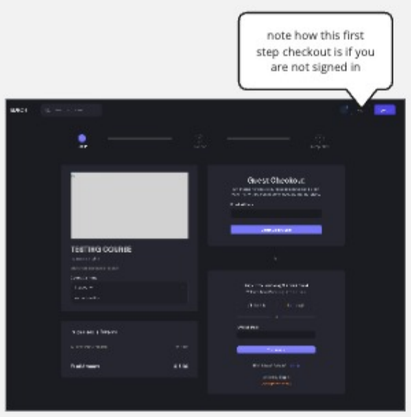
- We choose NoSQL because there is very few relationships
- Unless you need a lot of relationships like social media applications you don't need SQL
- SQL while it brings the ACID guarantees, but it also comes with a lot of overhead, maintenance, and in some cases scalability issues
- In our case we do not have many relationships and we can utilize nosql

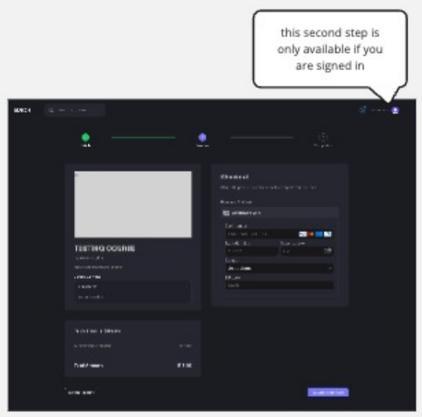
DynamoDB vs Other NoSQL

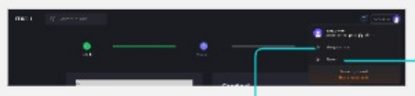
- Advantages
  - very fast and performant
  - fully managed compared to other databases like document db or mongodb, you don't need to handle infrastructure scaling etc
  - ideal when using AWS ecosystem and especially for serverless
  - generous **FREE** tier (always free up to an amount)
- Disadvantages
  - not ideal for tables with many relationships (but we don't have many here in this app)
  - not great for highly nested data structures
  - can be restrictive for complex filtering, sorting, aggregation

Auth Screen Flows



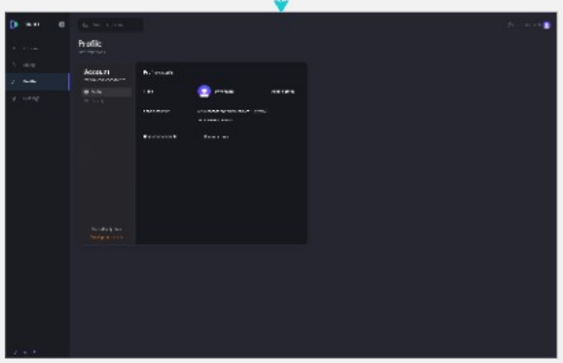


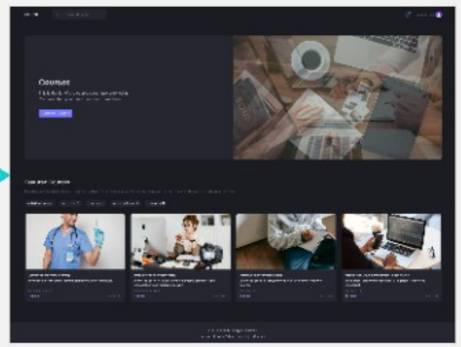




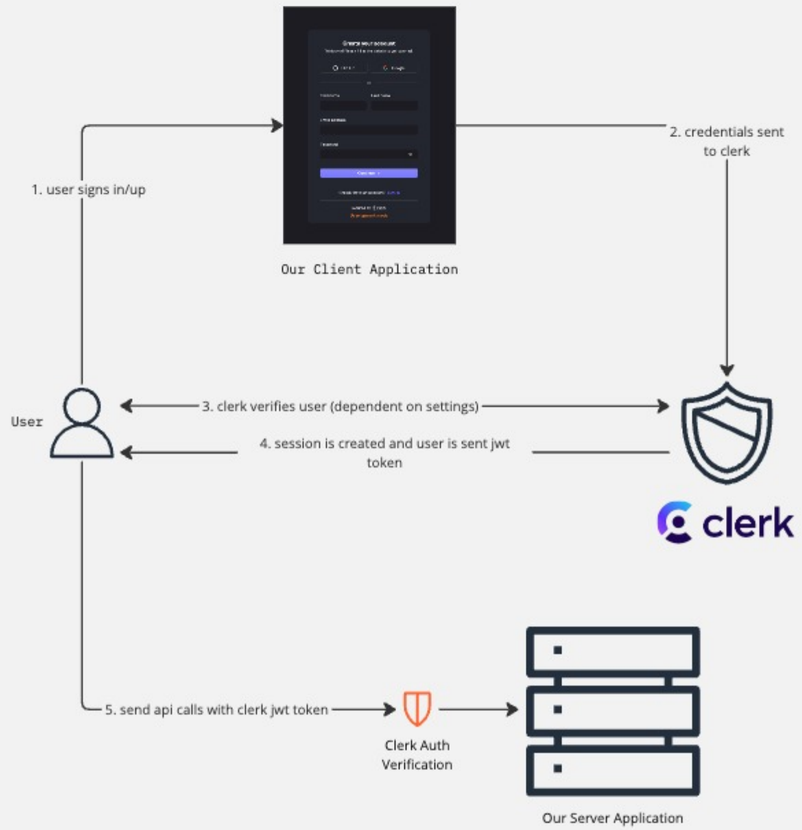
goes to either user or teacher profile page

goes to landing page





Authentication Diagram

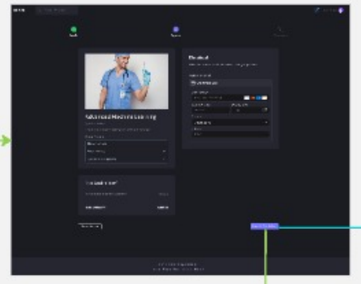


user hits this page if not signed in

user signs in/up

clerk

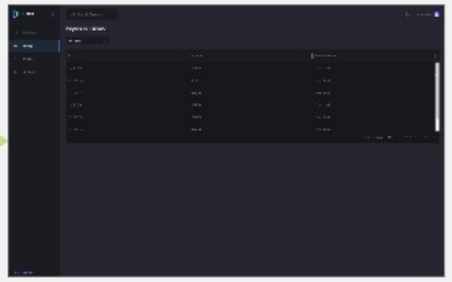
user hits this page if signed in



This creates transaction so user can fetch transaction records late

Transaction

Transaction information is used for the billing page. Also transaction stores stripe id so it can fetch the needed information from stripe.



This creates stripe payment

Stripe

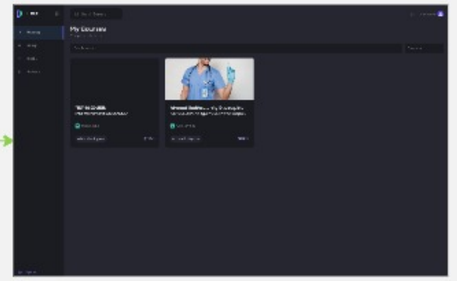
Need to add user enrollment to course

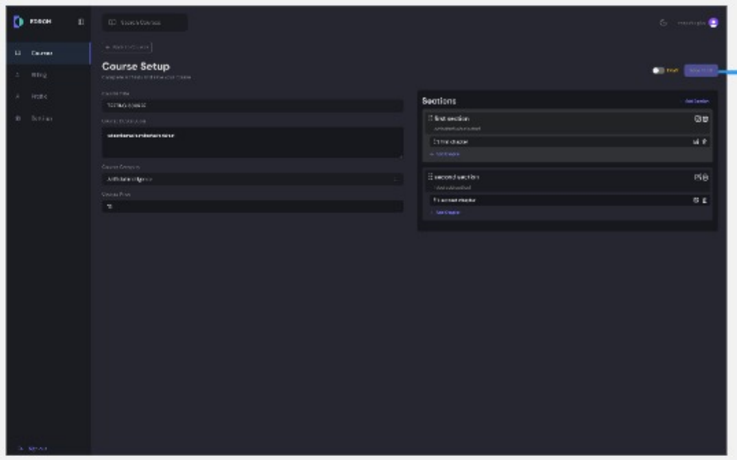
Course

This adds progress data for initial

UserCourseProgress

This is how we check if user is enrolled in a course





1. course data upload

Course

2. upload videos sequentially

S3

Amazon CloudFront

Chapter Modal Component

local state

on submit -
we push up local
state into redux
store sections

Section Modal Component

local state

on submit -
we push up local
state into redux
store sections

redux store
- sections

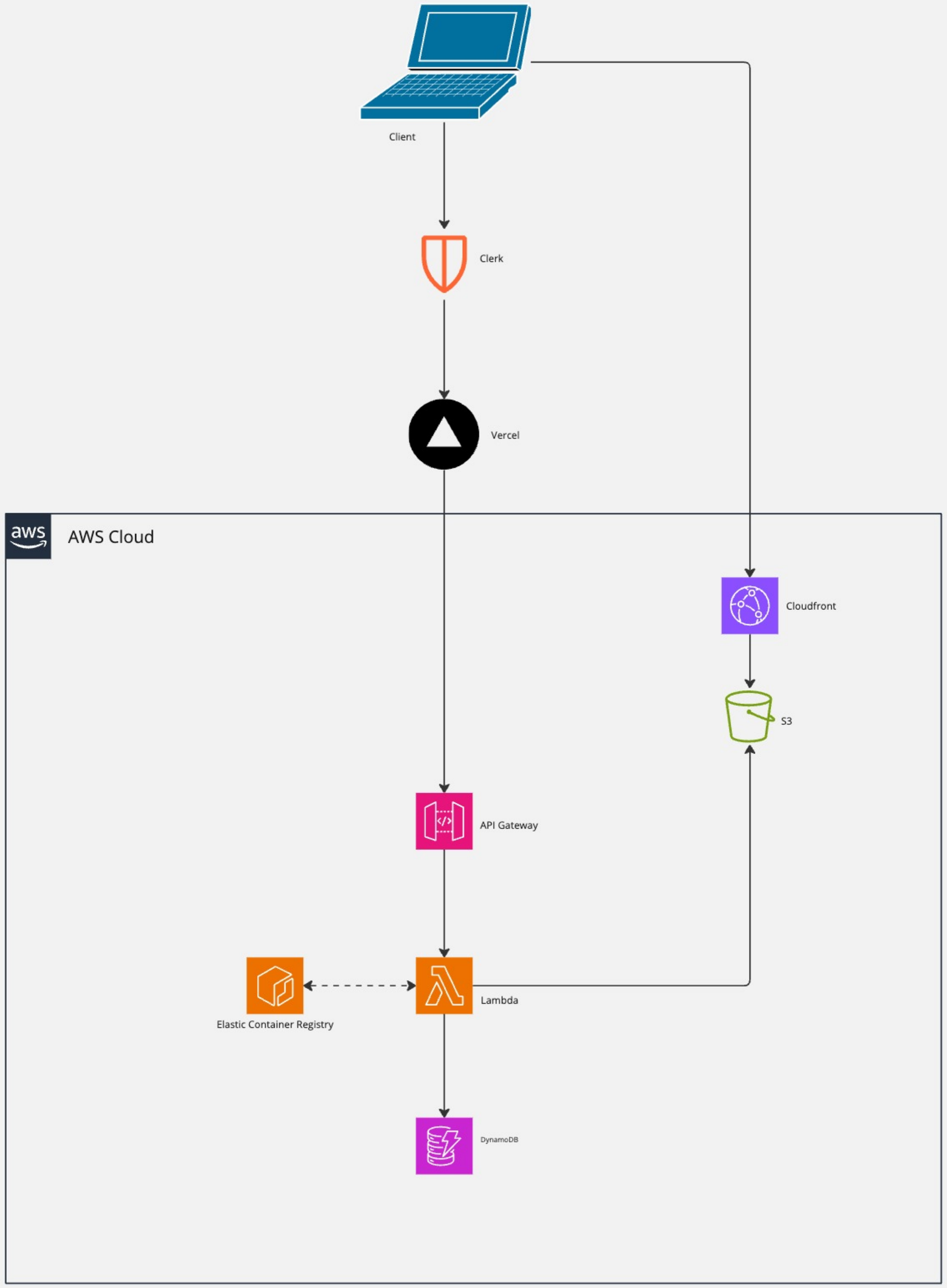
Course Editor Component

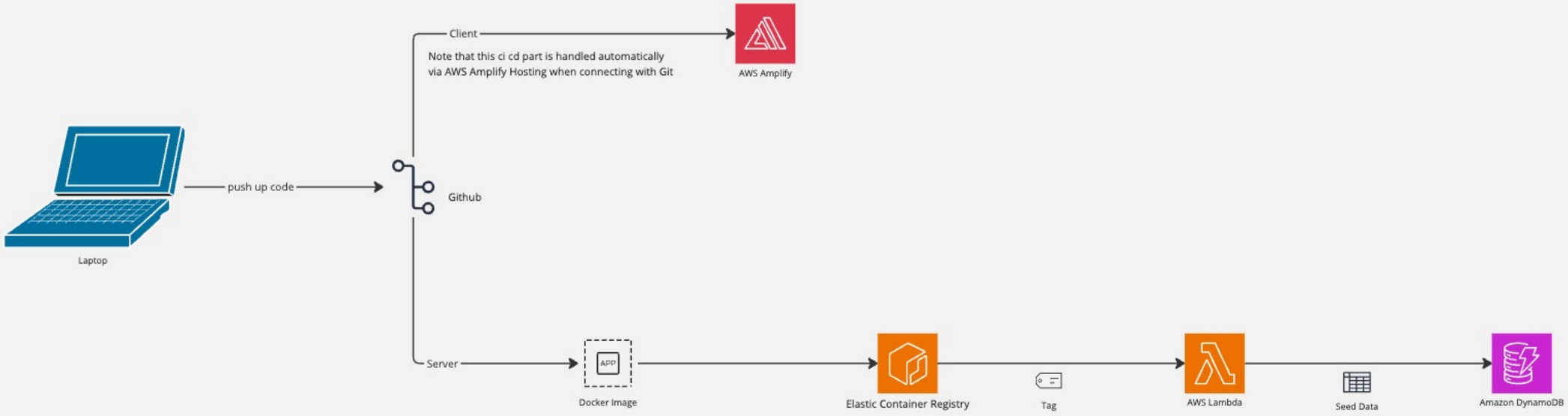
local state

+

redux store
- sections

when saving course, we use state from local and redux store


Client
Clerk
Vercel
aws
AWS Cloud
Cloudfront
S3
API Gateway
Lambda
Elastic Container Registry
DynamoDB


Client
Note that this ci cd part is handled automatically
via AWS Amplify Hosting when connecting with Git
AWS Amplify
push up code
Laptop
Github
Server
APP
Docker Image
Elastic Container Registry
Tag
AWS Lambda
Seed Data
Amazon DynamoDB