CRUDL Coding Challenge

The Task

In GoLang, build a simple CRUDL (Create, Read, Update, Delete, and List) web application that provides a JSON API for a collection of User objects.

- JSON API interface only — no front-end needed.
- A User record should have at the minimum an id, email, first & last name, and created/updated timestamp.
- Use a datastore of your choice, but it shouldn't require a standalone process. Some examples: sqlite, BadgerDB (Go)
- Provide a way for the app to be started with 10+ User records pre-populated. Some examples:
    1. Include with your submission a pre-populated DB file to be used when the app is run.
    2. Provide a data initialization function that can be run from the command line to populate the DB file before running the app.
    3. Define “upon start” logic in your app that populates the datastore when the app is run.
- Include a README file with the following information:
    1. How to run the application.
    2. An example cURL command to create a User record via the app’s JSON API.
    3. Approximately how much time you spent on the application.
    4. Any tradeoffs you made during development of the application.

Extra Credit

Add pagination to the list route based on “page=<page number>” and “per=<records per page>” query parameters.

Mega Extra Credit

In addition to the Extra Credit:

Allow pagination on the list route to be ordered based on “sort=<user attribute>” and “order=<asc or desc>” parameters.

Blow Us Away Option

In addition to the Extra Credit and Mega Extra Credit:

Add an RBAC aspect to the API so that only a User with an “admin” role can delete Users, and only a User with a “manager” or “admin” role can create/update Users.