CMPS 350 · Web Development Fundamentals

Lab 11 · Prisma ORM

Objective

- Prisma schema, migrate, and client for data modeling and access.

Resources

- Prisma Concepts: https://www.prisma.io/docs/concepts

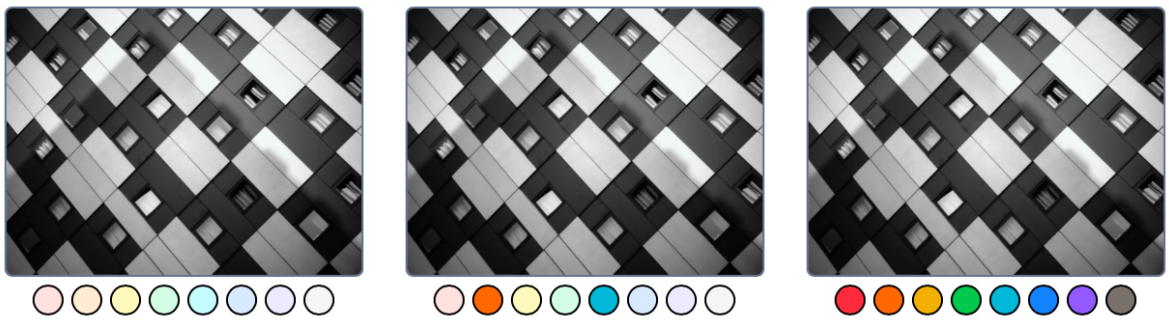
1. Project Task Tracker with Prisma

This exercise is about extending the project task tracker with an API and data repository service powered by Prisma.

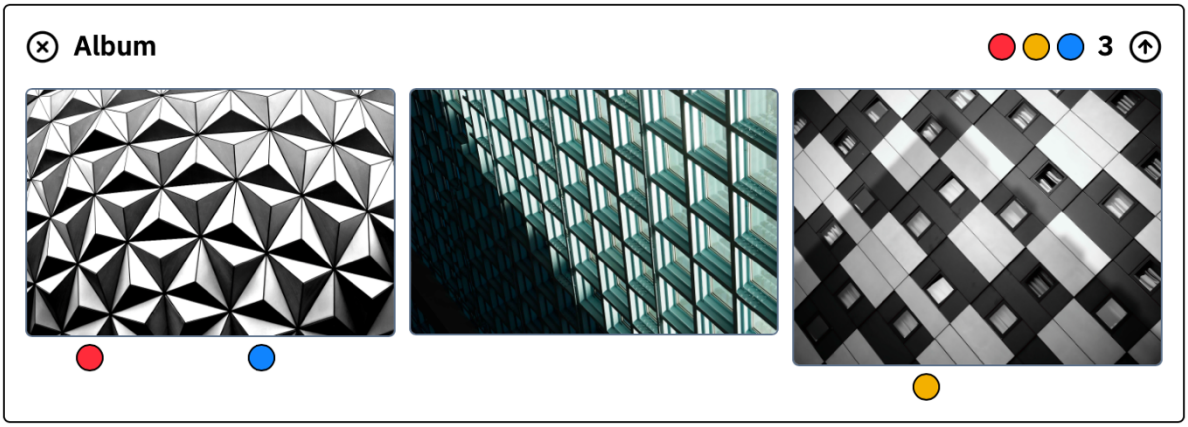
1. Define the data models using Prisma Schema.
2. Create the database using Prisma Migrate.
3. Create a data repository service with CRUD methods and use Prisma Client to access the data store.
4. Create a RESTful API using Next and use the data repository service in all routes.
5. Test the routes using Postman.
6. An interactive demo of this exercise is available at https://cmps350-project-task-tracker-prisma.vercel.app.

2. Photos Albums with Prisma

This exercise is about upgrading the photo albums data repository service to use Prisma and extending the data model.



1. Define the data models using Prisma Schema.
2. Extend the data models to include a fixed set of color tags for photos.
3. Create the database using Prisma Migrate.
4. Update the data repository service to use Prisma Client for data store access.
5. Test the routes using Postman.
6. Update the interface to display the tags.



7. Allow the user to filter the photos in an album by tag. The tags of an album are the union of its photo tags.
8. Use an effect and an event listener to allow the user to click on a photo and display it in full-window mode, then use the arrows keys to navigate between the photos in the same album, and finally close the full-window display by pressing the escape key or clicking on the photo itself.[1]
9. An interactive demo of this exercise is available at https://cmps350-photo-albums-prisma.vercel.app. Do no upload large files as the database size is capped to 500 MB.

3. Deployment using Vercel and Supabase

This exercise is about deploying a project using Vercel and Supabase.

1. Sign up with Supabase using GitHub: https://app.supabase.com.
2. Create a new project and follow this guide to setup the project using Prisma with Supabase: https://supabase.com/docs/guides/integrations/prisma.
   2.1. Create at least one table using Supabase before trying to connect to the database.
   2.2. Change the data source provider to PostgreSQL and set up connection pooling.
3. Add a `postinstall` script to `package.json` with `prisma generate`.
4. Sign up with Vercel using GitHub: https://vercel.com/signup.
5. Create a new Git repository with the project source and deploy it using Vercel.

[1] Photos from Pexels (https://www.pexels.com).