PERN Stack

Exercises

OVERVIEW

In this exercise you will be creating your first full stack application

GOALS

1. Initializing a node js project
2. Creating a postgres DB
3. Connecting the node APIÏ server to the db
4. Creating a react app that uses the API server

SPECIFICATIONS

In this project, we will be developing a full-stack PERN application that focuses on creating a library web application. The application will enable users to store new books and provide them with convenient search and filtering functionalities..

Exercises

Setup

- Create a github repository and clone it
- Initialize the package.json inside the root directory
- Create a new instance on elephantsql (or reset an existing instance)

Exercise 01

- Create a table called books, that has the following columns
    - Id
    - title
    - author
    - description
    - category
    - cover_url
    - publshedAt
    - isActive
- Create a basic express server
- Connect to the db using pg (use env variables for the db credentials)
- Create a react application inside of the repository
- Use the cors middleware to accept requests from the frontend

Key takeaways from Exercise 01:

- Setting Up the db, server, and frontend early protect us from time consuming refactoring in the future

Exercise 02

- Create an endpoint that accepts a POST request on path '/books' to save a new book in the database (test it using insomnia)
- In the frontend create a component that renders a from, and when the form is submitted send a POST request to the backend to add a new book to the database

Exercise 03

- Create an endpoint that accepts a GET request on path `/books` to retrieve all books
- Using query parameters filter the books and limit the amount of the retrieved books, for example `/books?limit=10&skip=0`
- Create an endpoint that accepts a DELETE request on path `/books/:id` that deletes a book with the matching id from the db
- In the frontend create a component that displays books from the db (without pagination)
    - Display the title, author, category, and the cover image for every book

Exercise 04

- Set up the routing in the frontend
    - Create a simple navbar that navigates the user to home, books
    - On path `/` display the home page (simple component that welcomes the user to the library app and has the form as well)
    - On path `/books` show the list of books
    - On path `/books/:id` show the details of the book with the matching id
- For every book in the books list display a view more button that navigates to the details page of one book
    - Show all the attributes of the book except the active status and id
    - Display a delete button that deletes the book

Exercise 05

- Implement pagination when displaying the books
- Implement filtration based on books category (backend filtration using query parameters)