
_nology
TALENT IN TECHNICOLOUR
Full-stack-project

Overview

You're going to build a full-stack web application!

This will solidify the concepts we've covered so far, built using these technologies:

- Java
- Spring
- MySQL
- React

Courseology

The application will have a front end, back end and will be able to **CRUD** a local database.

The project is based on having a selection of courses and students 😊.

If you are confident and have your own idea create it! Just make sure to cover the stages below. 😎

Setup

Create your Github Repository, inside you should have two folders.

- A React Frontend
  - `npx create-react-app courseology-frontend`
- A Java/SpringBoot Backend `https://start.spring.io`. Make sure you select the checkboxes and have the following dependencies.
  - Maven Project
  - Java v11
  - Jar
  - Spring Web
  - MySQL Driver
  - Spring Data JPA

Data Structure

I would advise you to think of your data structure first, think....

- What should a course have?
  - name, category, completionTime, price, syllabus, author etc...
- What should a student/user have?
  - name, email, interestedIn etc...
- You can use something like MURAL to plan out your structure and relationships.

You don't have to set up your database yet that will be covered in stage 3.

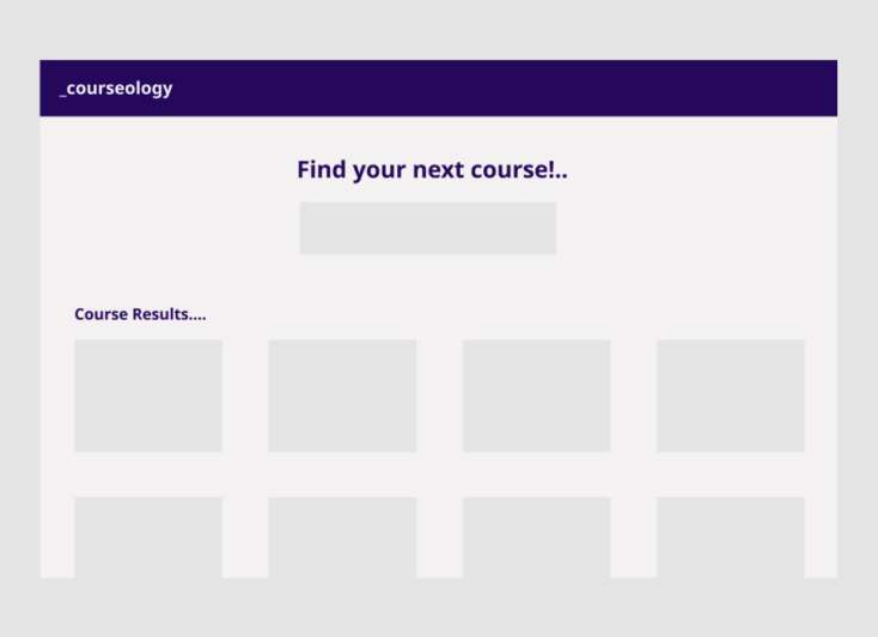
Stage 1:

Back end

Once you have the project setup and have an idea of the data you are modelling.

Create a controller called `CourseController` with a `/courses` endpoint with a which responds with all of your course data.

This doesn't have use a DB yet.



Front end

Create a courses page to show the courses fetched from your GET `/courses` endpoint.

It will have a searchbox to filter courses displayed.

Stage 2:

Back end

Create a endpoint for GET `courses/{courseId}` This should return the course with a matching id.

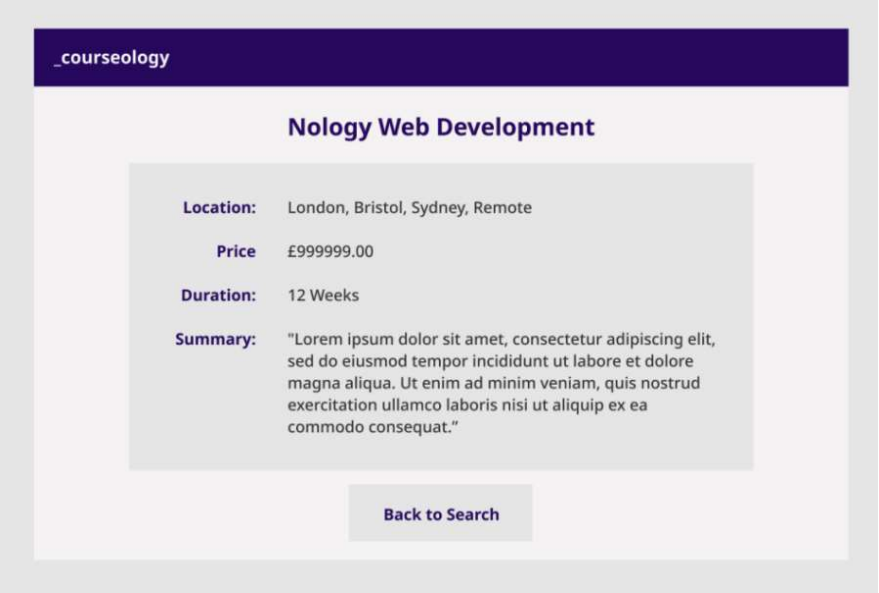
Front end

Create a new page for showing each individual courses based on their id.

When a user clicks on a course on the courses screen it should take them to a new page that displays all of the course details.

This should fetch the individual course data from your GET `courses/{courseId}` endpoint.

You will be able to go back to the page displaying all courses from here.



Stage 3:

Database

It is time to hook it all together create tables with relevant data in for courses and students.

- If you are writing SQL statements to create your data, keep a note of them.

Back end

Get your Spring application hooked up to your local DB. It should be querying it for information about the courses.

Front end

It will now display the courses that are stored in the DB.

Stage 4

Back end

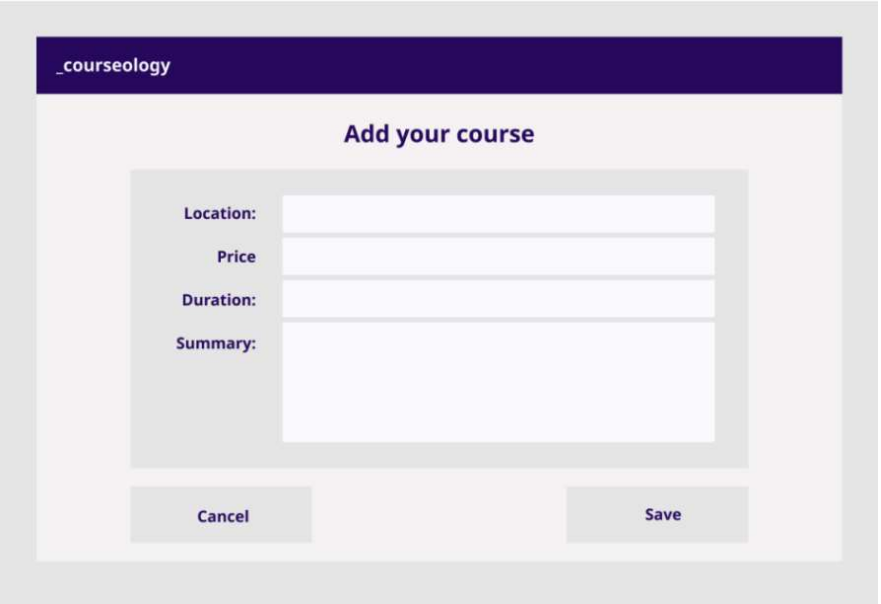
Create a endpoint for POST `courses` This should expect a `Course` in the body of the request. It should add it to the DB, it should return the newly created resource or resource Id.

Front end

Create a page for people to add their own courses. The form should send data to the POST `courses` endpoint. Once it has successfully added it it should navigate to the page with that courses individual information.

Database

Should store the newly created course in the correct table in the database.



Ready, Steady, Go!!!

We'll review how everyone is progressing with this challenge, but for now....

Good luck!