



TASK

Capstone Project - Full Stack with React and Express

Visit our website

Introduction

WELCOME TO THE FULL STACK APP WITH REACT AND EXPRESS CAPSTONE PROJECT!

Great job on making it to this point. In this Capstone Project, you will build a full-stack web application using React and Express that interfaces with the [iTunes Search API](#). The application should allow users to search for content within the iTunes Store and Apple Books Store and save a list of their favourites. This project will showcase your skills in both front- and back-end development, API integration, and basic documentation.

Instructions

This project involves creating an app that needs modules to run. These modules are located in a folder called 'node_modules', which is created when you run the command '`npm create-react-app my-app-name`' or similar. Please note that this folder contains hundreds of files which, if you're working directly from Dropbox, have the potential to slow down Dropbox sync and possibly your computer.

As a result, please follow this process when creating/running such apps:

- Create the app *on your local machine* (outside of Dropbox) by following the instructions in the task.
- When you're ready to have a reviewer review the app, please *delete the node_modules folder*.
- Compress the folder and upload it to Dropbox. Your reviewer will, in turn, decompress the folder, install the necessary modules and run the app from their local machine.

Capstone Project

Please consult the official documentation of the [iTunes Search API: Overview](#) to learn more about the iTunes Search API.

Carefully follow the instructions and ensure that your web application fulfils the following criteria:

Front-end:

- Create an attractive and user-friendly interface using React components.
- Implement a search input where users can enter a search term and also select the type of media they want to search for (e.g. movie, podcast, music, audiobook, short film, TV show, software, ebook, or all).
- Implement a favourites list where users can add and remove items. The list of favourite content does not need to be remembered when the user exits your web application.
- All output results should be attractively displayed.

Back-end:

- Use Node.js with the Express framework to create the server to serve your React front-end and handle API requests.
- Implement a route that communicates with the iTunes Search API to fetch music data based on user queries. Retrieve album information including album name, artist name, album cover image, and release date.
- Pass the API endpoints data to the front-end as needed to display results to the user.
- Use JWT to authorise API requests and secure your API.

Take Note:

- You do not have to authenticate users.
- You do not have to create a database for this application.
- No information about the users of your application, their favourites list, nor their history activity needs to be stored by your application after the user leaves the app.

Documentation:

- Create a **README.md** file that includes a description of your application, its purpose, and key features.
- Provide clear instructions on how a user can install and run your application locally.
- README files are compiled in Markdown, a lightweight markup language commonly used for project documentation. Markdown uses plain text with a special syntax allowing you to format elements such as headers, lists, links, and images. To create a README file, open Visual Studio Code or any preferred code editor, compose your content using [Markdown](#) syntax, and then save the file with a **.md** extension.
- **Optional:** Create wireframes using [draw.io](#) to demonstrate how the user interface of your application will work. Export the wireframes as a PDF and save it in your Dropbox folder as **wireframes.pdf**.

Guidelines:

- The file structure of the project should be well-organised and easy to understand and use.
- Your code should be well documented with appropriate comments. The code should also be easy to read, adhering to [Google's JavaScript Style Guide](#) about indentation, meaningful variables, component names, etc.
- You should ensure that your application has been appropriately tested and is free from errors or bugs. It's recommended to try using unit testing to help with this process.
- A code reviewer should be able to launch your application by typing '**npm start**' from the command line interface.

OPTIONAL

- Deploy your application on Render and add the link to your application in the README.md file of your project.
- Please note that you can only deploy static apps, hosted on GitHub, to Render. In order to deploy a dynamic application, like this capstone project app, you require a paid Render membership.



Rate us

Share your thoughts

HyperionDev strives to provide internationally-excellent course content that helps you achieve your learning outcomes.

Think that the content of this task, or this course as a whole, can be improved, or think we've done a good job?

[Click here](#) to share your thoughts anonymously.

