

Goals

The goal of this project is to get an understanding of your backend skills. Please do not spend more than 4 hours on this assignment. You can use any language and/or framework that you would like. At the end of the test please send us a zip of your project.

Summary

We're going to create a simple project which will take in jokes from an API, store them in a database, and then allow users to search for a joke as well as rate and comment on a joke. Don't worry about the UI piece for now, just create a functional HTML/CSS layout which accesses your API. If you're comfortable with a CSS framework like Bootstrap or Foundation feel free to use that as a base.

Components

The Jokes

Please look over the documentation on <https://icanhazdadjoke.com/api>, specifically the "Search for Jokes API". Please create a database with tables that will store the results of a search against the API. Keep in mind that new jokes may be added to the API at any time. For this project we want to include any joke that is returned when searching for the word "hey", please use the JSON version of the API they provide. We will want to be able to run the import script as often as we'd like so we can get any new awesome jokes that may be added.

Please be sure to include in your project's README.md how to setup and run your joke importer.

The User Search Form.

Please build a form which allows a person to search our database of jokes. Whenever they search for term(s) it should look for the exact search they are looking for in either the joke itself OR the joke's ID (as defined by the API). This should **not** use the ICanHazDadJoke API's search functionality but you should build the logic to search and return the jokes from our own API.

Joke Ratings

When a user finds a joke they like please have it show only the joke they have selected.

From there allow the user to rate the joke on a scale of 1 to 5.

You should calculate and display the joke's average rating which is defined as taking the total rating value/# of ratings.

Joke Comments

Allow a user to leave on any joke a comment. Let the user enter whatever name they want the comment to

associated with.

Then on the page which shows a specific joke display the comments.

Pages

You'll need to create two "views" which include the functionality described above.

Search Page

A view where you search for jokes using the "User Search Form" described above. Include the rating bar and average ratings here.

View Joke

A view where you see the details for a specific joke. Place the comments functionality described above on this page.