Take Home Option 1 - LotsOfVids

1. Write a python function that takes a video and saves it to disk

2. Add testing to your function

3. Improve the service provided at the GitHub link by making it efficient, and log/monitor the performance.

Basically, the setup gives you a mostly working (but not very well) “camera-like” service that POSTs videos, at alarming speed, to a flask dev application that receives the POST. It’s all wired up. The challenge is to:
i) make it save the videos, and persist them when the container exits and
ii) make it do so at as high a rate as possible, without sacrificing write reliability

Please read the directions in the included README.md carefully for more details!

Take Home - Option 2

1. Write a python function that takes some data structure, for instance users and a list of books borrowed from a library and stores them into a DB

2. Make sure the function works for a sample dataset (please create your own, or grab one from https://www.kaggle.com/datasets)

3. Build frontend with Modern Front End JS Library - don't worry about the style and the look - which allows a user to borrow books from the library:
    a. Design a schema from data (suggestion: Sqlite/MySQL/PG)
    b. Create API to interact with database (suggestion: falcon, flask, express)
    c. UI Should have an input field - say the book title - with autocomplete, displaying the entries from the DB through the API.
    d. * On Selection of search result, save as "Selected" (borrowed) to DB.
        i. Bonus points: connect book to user
    e. UI Should have a table showing selected entries:
        i. Column 1: <Property of Data> 1
        ii. Column 2: <Property of Data> 2
        iii. Column 3: Delete button (returned)
    f. delete button should unselect <entry> from DB.

    g. Add a test suite to your implementation
    h. Include a README on how to run your solution.

4. Bonus points: package the service in Docker/docker-compose for easy deployment