SDE-2 - File Upload

Objective

You are working as an engineer for **IMDB** the world's most popular and authoritative source for movie, TV and celebrity content.

Requirements

You are tasked to build a feature where the content team can upload data related to movies using a CSV.

Functional

1. The user should be able to login to dashboard
2. Logged In users should be able upload CSV (format of CSV attached below)
3. Logged In users should be able to view the progress of their uploaded CSV
4. Logged In users should view the list of all movies/shows available in the system in a paginated view with functionality to sort items based on Date Added, Release Date, Duration

Technical

Scale

- Max CSV size can be 10 GB with billions of rows

Tech Stack

- Flask
- PyMongo for MongoDB

Requirement -

1. APIs for authentication
2. APIs to upload the CSV
3. APIs to track the status of uploaded CSV progress
4. APIs to view uploaded data in the dashboard with pagination and ability to sort items based on Date Added, Release Date, Duration

> 💡 Please create a basic, functional UI to demonstrate the core capabilities of each API. Note that UI design is not a focus of this evaluation.

Sample CSV

movies https://docs.google.com/spreadsheets/d/1S7oGjQXFIybfyKpj1rFWNRtjGcC7n_eQy0TM6nsLFRw/edit?usp=drivesdk

Submission Guidelines

1. Share your code via a public repository (GitHub, Bitbucket, etc.).
2. Include a README file with instructions on how to set up and run the system.
3. Provide a JSON file describing the database schema and indexes on each collection

```
{
    "<collection_name>": {
        "indexes": [],
        "schema": {
            "<field_name>": {
                "type": <int/float/str/list/dict>
            }
        }
    },
    ...
}
```

Evaluation Criteria

- Code structure and organisation
- Adherence to best practices in coding and database design
- Functionality implementation
- Documentation completeness and clarity

- Test Cases