Project Title	Web Scrapping Application
Technologies	MERN
Domain	Education
Project Level	Hard

Table

Table

1. Problem Statement:

Design a web scrapping application which can scrap the data of YouTube channel videos and will store them in a JSON file.

1.1. Overview of Web Scrapping Application?

Web scraping is a method for utilizing a script to scrape data from websites. Using web scraping, the time-consuming task of copying data from numerous websites can be automated.

When attractive websites don't expose an API to retrieve the data, web scraping is typically used.

The challenge presented to you in this situation is that you must collect the video's title, description, number of views, and upload date from the provided video link. The user will provide the YouTube video URL, and you must gather the data, save it in a JSON file, and then show it on the website.

When a user shares a channel link with you, you must collect all of the channel's video data and store it in JSON in order to display it.

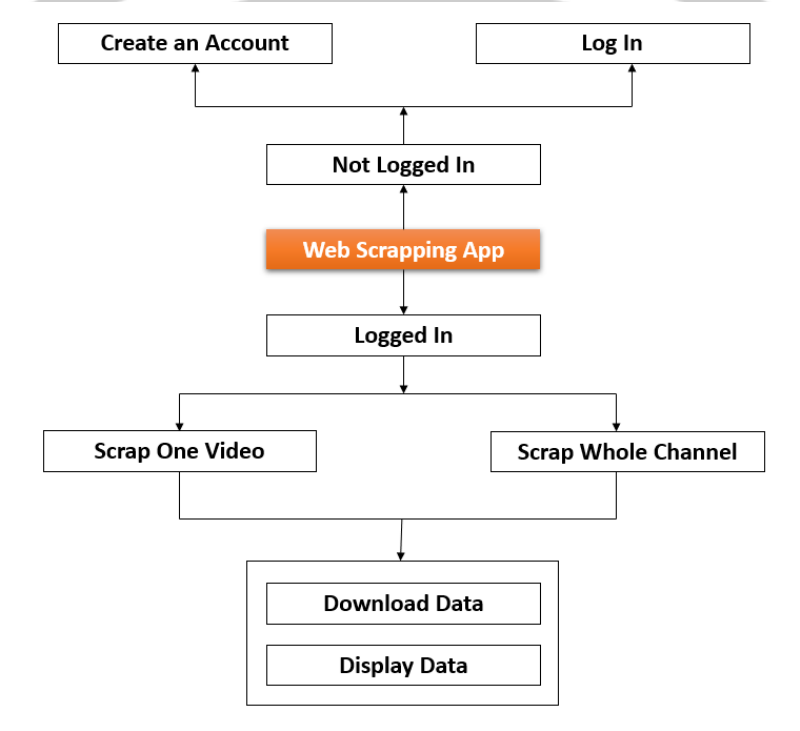
1.1.1 Features to be included in the Web Scraping Application

Here is a comprehensive list of the qualities that your website for web scraping should have-

a. The user will land on the website and will have option to login or create an account without authentication he will not be able to collect / scrap data from YouTube.
b. A logged in user can scrap the data from the YouTube.
c. The user can scrap a single video or can scrap the whole YouTube channel videos at once by providing the link.

d. After collecting the data from the website, it will be stored in a file (JSON) which can be downloaded from the website and can also be viewed there directly.
e. On the user dashboard he will be able to see all his scrapped details including the file and its history that what and when he had searched and scraped the data.
f. Also, he will be able to edit his profile details like name, password etc.
g. The user interface to display the data should be user friendly and attractive.
h. The Website should be completely responsive for all the screen sizes.

Low Level user flow



1.2. Project Objective

• **Web Scrapping:** A straightforward and user-friendly application that allows users to view all the details of a YouTube video or a channel's videos with a single click.

• **User-friendliness:** The project should have a simple user interface and be straightforward to use.

1.3. Scope of The Project

1. Gathering meta data from the YouTube gives us an edge to better our content, such as the title, description, length, and upload date.

2. A web application with a basic, intuitive user interface that is easy to use.

3. The user can learn more about other YouTubers to get a better idea of his own future content.

1.4. Functional and Non-Functional Requirements: -

1.4.1. Functional Requirements

1. **User Registration:** The application must allow users to sign up using their email, username, and password. Users must be able to self-register or, if they already have an account, log in immediately after opening the application. If the user disregards this phase. The user will not be able to scrape the data from YouTube if they omit this step.

2. **Saving:** The website should enable users to store their scraped information to their profiles so they may access or download it later.

3. **Edit Profile:** In the future, the logged-in user should have the option to change his profile information.

5. **Edit Scrapped Data:** The user should be able to examine and delete the data that has been scraped from his dashboard.

1.4.2. Non-Functional Requirements

1. **Privacy:** Only users who are logged in can conduct CRUD operations on both their scraped data and their profile data.

2. **Robustness:** To enable recovery in the event that the user's system fails, a backup of the scrapped data must be kept on distant database servers.

3. **Performance:** The application must load the landing page quickly and be lightweight.

1.5. Use Case Table

Authentication System	Register, Login, Logout	User
User Dashboard	Display User’s details and Scrapped Data	User
User Profile	Edit and Delete the Scrapped Data	User

Table 1. Use Case

2. Project Evaluation Metrics:

2.1. Code:

- You are supposed to write code in a modular fashion
- Safe: It can be used without causing harm.
- Testable: It can be tested at the code level.
- Maintainable: It can be maintained, even as your codebase grows.
- Portable: It works the same in every environment (operating system).
- You have to maintain your code on GitHub.

- You have to keep your GitHub repo public so that anyone can check your code.
- Proper readme file you have to maintain for any project development.
- You should include the basic workflow and execution of the entire project in the readme file on GitHub.
- Follow the coding standards.

2.2. Database:

MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas.

2.3. API Details or User Interface:

You have to expose your complete solution as an API or try to create a user interface for your model testing. Anything will be fine for us.

2.4. Deployment:

Implementation of reverse proxy, load balancing, and security group is mandatory for deployed applications.

2.5. Solutions Design:

You have to submit complete solution design strategies in High-level Document (HLD), Low-level Document (LLD), and Wireframe documents**.**

2.6. System Architecture:

You have to submit a system architecture design in your wireframe document and architecture document.

2.7. Optimization of solutions:

Try to optimize your solution on code level, architecture level, and mention all of these things in your final submission.

Mention your test cases for your project.

3. Submission requirements:

3.1. High-level Document:

You have to create a high-level document design for your project. You can reference the HLD form below the link.

Sample link: HLD Document Link

3.2. Low-level document:

You have to create a Low-level document design for your project; you can refer to the LLD from the link below.

Sample link: LLD Document Link

3.3. Architecture:

You have to create an Architecture document design for your project; you can refer to the Architecture from the link below.

Sample link: Architecture sample link

3.4. Wireframe:

You have to create a Wireframe document design for your project; refer to the Wireframe from the link below.

Demo link: Wireframe Document Link

3.5. Project code:

You have to submit your code to the GitHub repo in your dashboard when the final submission of your project.

Demo link: Project code sample link

3.6. Detail project report:

You have to create a detailed project report and submit that document as per the given sample.

Demo link: DPR sample link

3.7. Project demo video:

You have to record a project demo video for at least 5 Minutes and submit that link as per the given demo.

3.8. The project LinkedIn a post:

You have to post your project details on LinkedIn and submit that post link in your dashboard in your respective field.