Page1:

1. \*\*Unique Access Tokens:\*\* Generate unique access tokens for each user. These tokens can be used to track and limit the numberof times a user schedules a Calendly link or accesses video content.

Example:

* Generate a unique token when a user is created
* Save the token along with the user details in a database
* Track how many times a token is used
* This token Expiery after 7 days 8 days.

1. \*\*Membership Levels:\*\* Create different membership levels with specific access rights. Users can be assigned to these levels based on their usage, and their access can be restricted accordingly.

Example:

* Save membership method in the database.
* Membership Method Free (14) , Basic (30), Pro(60), VIP(90) this method save in database and user first time use free trial.
* After 14 days this method expire and user update the membership Free trial only one time
* Use Stripe/Paypal payment integration.

1. \*\*Database Tracking:\*\* Use your platform's database to track user activities. Store information about when a user schedules aCalendly link or accesses a video, and implement logic to prevent repeat access.

Example:

* User results save in database with IP, User\_id, current date etc.
* Any module update after that save in log activity so user view the specific Module activity regarding, calendly link or accesses video
* User has already performed the action before allowing access.
* Already accessed than the user is blocked from repeating the action.

4. \*\*API Integration:\*\* If you have technical expertise, integrate with Calendly's API to track scheduling activities and enforce access limits. You can also use video platform APIs (such as Vimeo, YouTube, etc.) to track and control video access.

Example:

* Get user data from calendly and save in database.
* Video view using API.
* Youtube Api implement in this.
* Video views using Api.
* Same as Vimeos
* **Users are restricted** from repeating the same action after accessing content

1. \*\*Expiry Mechanism:\*\* Implement time-based access controls. For example, a user can only access the video programming within a certain time window after they first log in.

Example:

* It also sets the **expiry duration (ex. 24 hours)**
* **Check if the time has expired than user in denied.**
* **Than valid the watch the video.**

6. \*\*User Authentication:\*\* Ensure that users must authenticate themselves before accessing any resources. Combine this with themethods above to enforce access restrictions based on their unique identifiers.

Example:

* If user login detail not match than redirect login page.
* If a user **exceeds the expiry time**, access is denied.

I Think the best Option is 1 and 2 Because

1. Generate Unique token for every users and set time limit and handle easy way.
2. User every detail check according to Membership days.

Page2:

1. Language independent (Mandarin, English, etc….). All information should be in the local language + translated into English in the backend/due diligence document.

Example:

* Every text should be defined in the language file for all pages.
* Users can **switch languages** (e.g., English, Mandarin).
* This language save in Database and change According to this

Please discuss how to implement the other four tasks and their uses.