Backend Design Document

Home Page

The 3 file system:
The content in s3 bucket will be split into three parts:

- One file which stores all the list of festivals and their details. This is static and will be permanently persisted. Only updated in case new festivals are to be included.
- A second file which will provide the data of festivals for the current day. The filename will be the date it corresponds to in DDMMYYYY format.
- A third file which will contain a list of data of the previous day(reason for maintaining a 3rd file further explained). The filename will also be the date it corresponds to in DDMMYYYY format.

The 3 file system will avoid the need of picking the relevant data to be showcased everyday by the FE. This will reduce the render time of the home page.

The 2nd and 3rd file will be updated everyday by a cron job at 12 midday.This can be directly read by FE and then further operations can be performed on it.

Both current and previous day files are maintained for precaution in case the website is accessed at around midnight, which might lead the website to pull data either from 2nd or 3rd file in a matter of minutes

The Home Page Cron Job:
Frequency: once in 24hrs
Time: 12:00 p.m.(Midday)
Task: Delete the already existing previous day file and create a new file for the next day.
Ex:
On 5th Jan it will have these two files before 12pm.:

- 04012023
- 05012023

After 12pm., The old file belonging to 4th Jan(previous day) will be deleted and a new file corresponding to 6th Jan(next day) will be created:

- 05012023
- 06012023

Expectation: FE will use the date of request to fetch the corresponding file as needed.

Note: Filtering, sorting and searching will have to be handled in FE side

Templates Page

1) Api to show all templates corresponding to the festival chosen.
   API endpoint: /templates
   Type: GET
   Request Params:
   - Query Param:
     - festival_id(string): ID of the festival chosen in home page

   Response:
   - templates(List[TemplateObjects]): Each TemplateObject contains:
     - template_id:(String)
     - icon:(byte String)

Designing Wish Pages:

1) API to get list of images for each festival:
   API Endpoint: /images/show
   Type: GET
   Request Params:
   - Query Param:
     - festival_id(string): ID of the festival chosen in home page

   Response:
   - images(List[ImageObject]): Each ImageObject contains:
     - image_id:(String)
     - image_data:(byte String)

2) API to upload image:
   API Endpoint: /images/upload
   Type: POST
   Request Params:
   - Body Param:
     - image_data(byte String): Binary data of the image uploaded by the user

   Response:
   - image_id(String): Newly generated ID for the string

Final Creation Page:

1) API to hit when create button is clicked
   API Endpoint: /wish/create
   Type: POST
   Request Params:
   - Body Params:
     - animation_id(String)
     - image_id(String)
     - message(String)
     - party_effects_id(String)
   - Response:
     - url(String): Newly generated URL which can be shared with others