Report on System Limitations, Recommendations and Structure

Date: 1/2/2025
Prepared by: Arooba Zaman

---

1. Executive Summary

This report provides a detailed overview of the system's limitations and practical recommendations for optimizing usage. The system integrates the Printify API, Instagram API, and Facebook API to automate product creation and posting processes.

---

2. Scope of the Report

This report covers:

- The limitations of the current implementation.
- Recommendations for overcoming these limitations.
- Best practices for using the system effectively.

---

3. System Limitations

3.1 Design Limitations

Text Field Restrictions:
For Design 1, the text_1 field performs optimally with around 11 characters. Texts exceeding this length may not yield the best design output or visual results on the product.

Recommendation: Ensure that text_1 is limited to 11 characters to achieve optimal results. If longer text is necessary, consider adjusting the design parameters to accommodate it.

3.2 API Limitations

Printify API Refresh:
The Printify API does not support token refreshing, and the API connection is expected to remain static. This limitation may cause issues when the connection needs to be updated.

Recommendation: The Printify API token should be reviewed and changed periodically, ideally once a month, to avoid potential connection issues. A scheduled task to update the token could be implemented.

---

4. Impact of Limitations

The limitations mentioned above can have the following impacts:

Design Quality: Longer texts may result in poor design rendering or misalignment in the printed product.

Printify Token: If not updated periodically, the Printify API token could become stale, causing failed product creation or issues with image uploads.

---

5. Recommendations for Optimal Use

5.1 Workarounds

Text Length Adjustment: Use a character limit validator on the front end when users input text for Design 1. If the text exceeds the optimal character count, provide a warning or auto-trim it.

API Token Management: Automate the process of reviewing and updating the Printify API token monthly. Use a background service or scheduled task to check for token validity and refresh if necessary.

5.2 Best Practices

Regular Monitoring: Regularly monitor the system's performance and ensure that tokens are updated and database entries are accurate.

User Input Restrictions: For Design 1, provide guidelines for the text field to limit input to a maximum of 11 characters. This will help avoid design issues and maintain consistency.

---

6. MongoDB Schema

Field	Type	Description
_id	String	Auto-generated unique ID
design_id	String	Unique ID for the design to be selected **(1, 2, 3)**
text_1	String	Main required text for the design **(recommended up to 11 characters for Design 1).**
text_2	String	Optional text field **(only for Design 1)**
product_title	String	Optional, for future implementation **but mandatory in schema**. just include empty string.

product_description	String	Detailed description of the product **(e.g., A high-quality t-shirt...).**
date_time	Date	Scheduled date and time for Instagram posting **(e.g., 2024-10-30T10:00:00).**
caption	String	Caption to be posted with the Instagram image.
status	String	Current status of the schedule**:(pending for initial)**
images	Array	List of product images generated and stored.
tags	Array	Tags associated with the product
to_delete	String	Initially empty. Populated with image URLs after fetching from Printify.
createdAt	Date	Optional to keep track of entry creation **but can keep dummy date**
updatedAt	Date	Optional to keep track of update entry **but can keep dummy date**

---

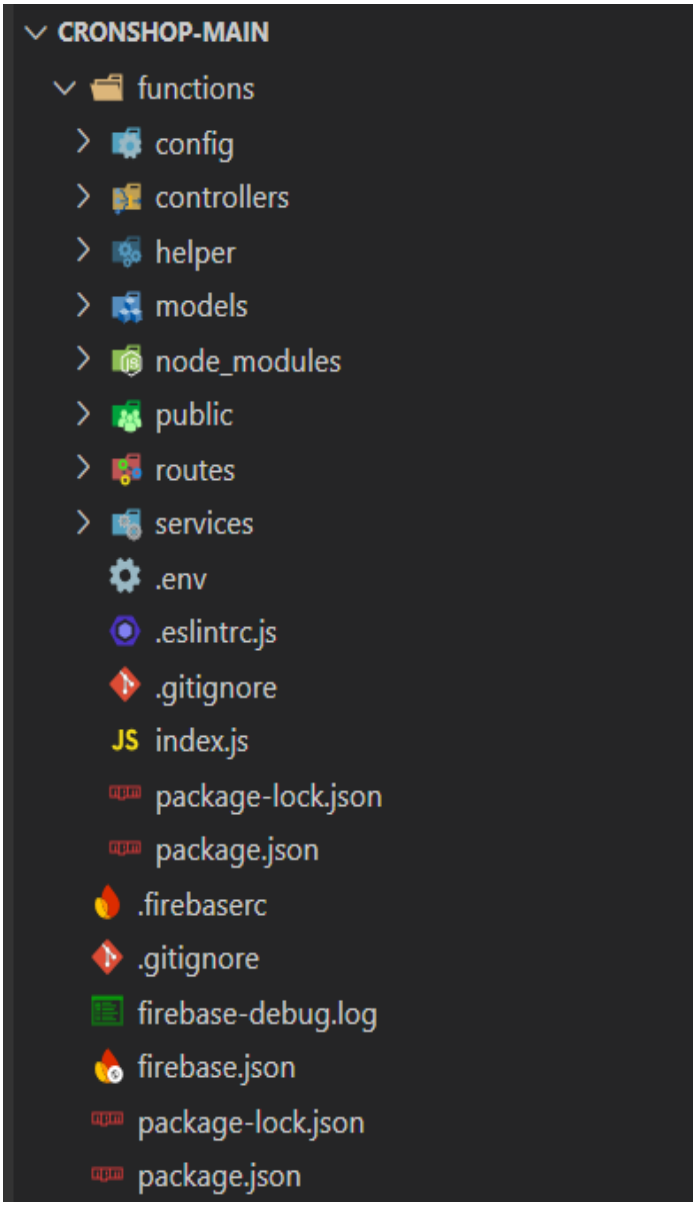
7. Example Schema that can be modified and used

```
{
  "_id": {
    "$oid": "67763687dafc4e2e7e2ac815"
  },
  "design_id": " 1 ",                                  // Here you can use 1, 2, 3
  "text_1": "Hi I am Nat",                             // Text for all three designs but nym text for design 1
  "text_2": "short for natural disaster",              // Definition text for design 1 only
  "product_title": "Awesome T-shirt",                  // Not Getting used for now keep as it is
  "product_description": [                             // Printify captions for candle, mug, tshirt and hoodie
    "NAT Candle",
    "NAT Mug",
    "NAT tshirt",
    "NAT Hoodie"
  ],
  "date_time": 2024-10-30T10:00:00Z,                   // Change as per the time you want to post on insta (utc)
  "caption": "Meet Nat – short for Natural Disaster!", // Caption for Instagram post
  "status": "pending",                                 // Keep it as it is
  "printifyProductId": "",                             // Keep it as it is
```

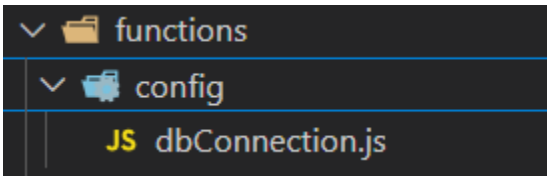
```
  "images": [],                                  // Keep it as it is
  "tags": [],                                    // Tags for Shopify and Printify products
  "to_delete": "",                               // Keep it as it is
  "createdAt": 2024-10-22T12:00:00Z,             // Keep it as it is  (Date type : utc)
  "updatedAt": 2024-10-22T12:00:00Z              // Keep it as it is  (Date type : utc)
}
```

- Things that are to be kept same
- Compulsory stuff to be changed as per need
- Optional Stuff for some posts

8. Project File Structure

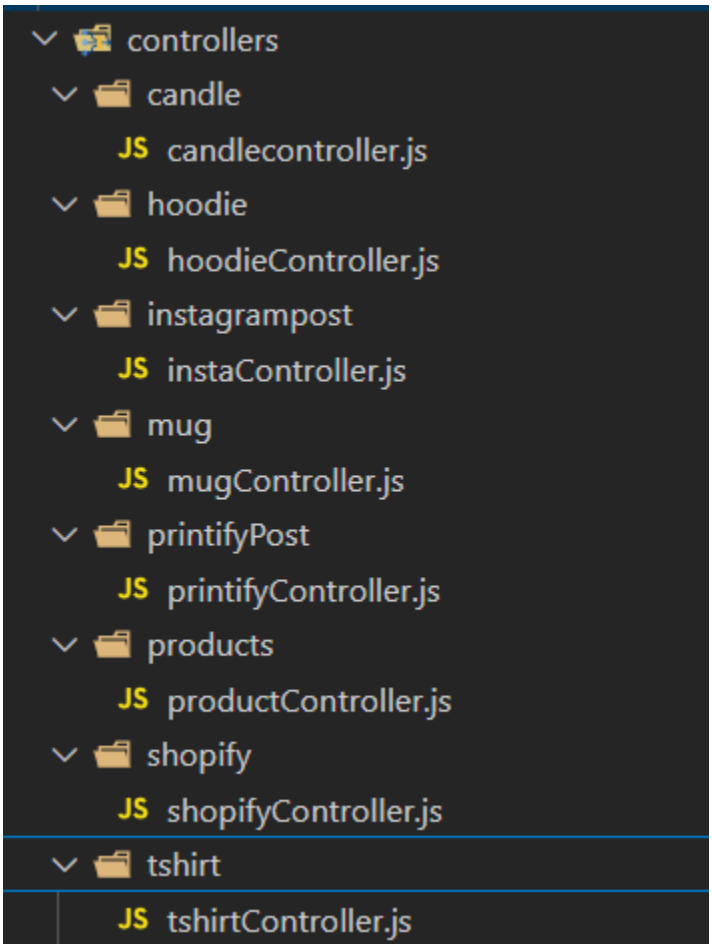


8.1 Config



File for database integration code that takes the connection string of Mongodb for connectivity.

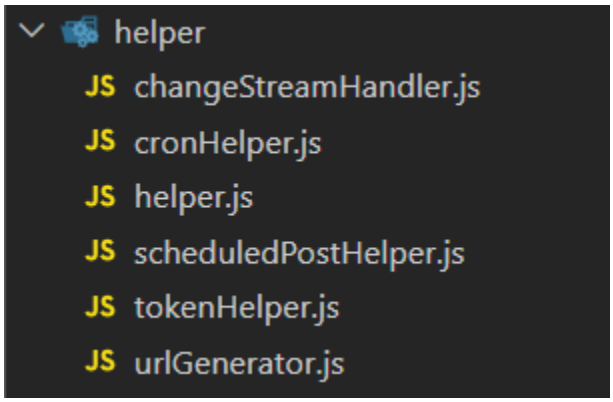
8.2 Controller



- **Controller** has all the business logic including the product creation, posting on Instagram, publishing on shopify and interactions with database.
- **Candle Controller** has the logic to create design and candle product on Printify as well as publish it. After creating it returns the image url of product and product id.
- **T-shirt Controller** has the logic to create design and shirt product on Printify as well as publish it. After creating it returns the image url of product and product id.
- **Hoodie Controller** has the logic to create design and hoodie product on Printify as well as publish it. After creating it returns the image url of product and product id.

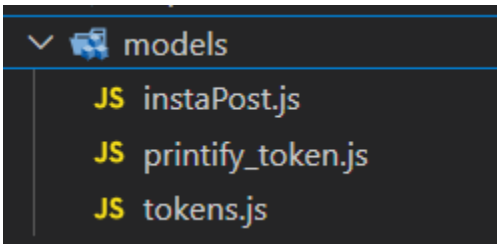
- **Mug Controller** has the logic to create design and mug product on Printify as well as publish it. After creating it returns the image url of product and product id.
- **Instagram Post Controller** has the logic to get the image urls of all products, along with the caption from database and make a carousel post on Instagram. It also checks through a helper function that if the token for insta api is expired or still in working state.
- **Printify Post Controller** has the logic to post images of design to **printify** in order to put the design image on products.
- **Shopify Controller** has the logic to post the products on shopify.
- **Products Controller** has the logic to use all the above controllers of the product creation and post all the image urls of products to database.

8.3 Helper functions



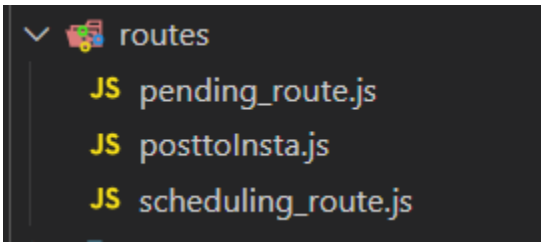
- The helper folder to store reusable code that performs common tasks.
- **changeStreamHandler** is called whenever the pending post route is accessed that checks for posts in the database with pending status and in case of any error thrown, the necessary operations happen to make the process robust.
- **cronHelper** has the logic to call the function to check the token of Instagram rather expired and if yes, call the function to refresh it.
- **helper** has the functions
    1. to convert image into base64
    2. to get random names for images to save them distinctly
    3. to get random colored products for Instagram posting
    4. to filter certain types of designs types of mockups
- **scheduledPostHelper** has the logic to check for the posts from database to check if the post time has reached then change the status to posting and then after getting successful posting, delete that product from database.
- **tokenHelper** has the logic to refresh the token of Instagram using facebook api.
- **urlGenerator** has the logic to post the thumbnail images for insta post to google drive and get live urls.

8.4 Models



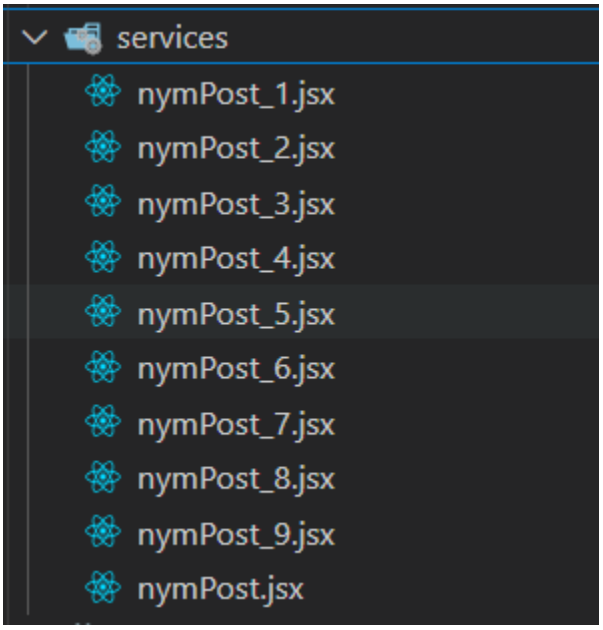
- The models folder has the schemas that are according to database collections.

8.5 Routes



- These are the routes that are getting hit according to crons timings
- The pending route is for making products
- Scheduling route is for posting scheduled posts on insta

8.6 Services



- This file has the design code for all types of products, variants and the thumbnails .

9. Deployment Settings

9.1 Firebase Login

https://youtu.be/gptBM2CPMQs?si=ifKSuDj9RpcXcraF

https://firebase.google.com/docs/cli#install_the_firebase_cli

9.2 Commands

9.2.1 To Run in local environment

Run this command in terminal of project directory outside the functions folder to make changes in code and test in local environment.

firebase emulators:start --only functions

9.2.2 To Redeploy

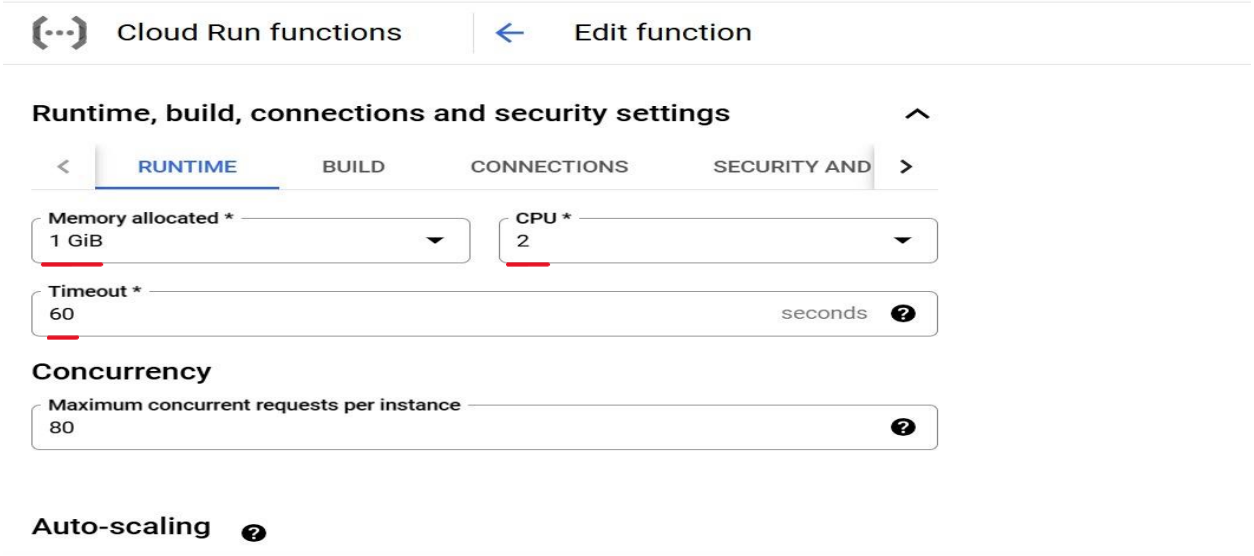
Run this command in terminal of project directory outside the functions folder

firebase deploy

9.2.3 Resetting cpu, memory usage

After redeployment, visit this url and the settings should match these to avoid extra payments.

https://console.cloud.google.com/functions/details/us-central1/api?env=gen2&invt=AbltQg&project=whimnym-on-fire



9.2.4 Check for Project logs

Go to this url to check for project logs

https://console.firebase.google.com/project/whimnym-on-fire/functions