Comp2Dust Project Overview

1.Description

Project **Comp2Dust**

Prepared by Yiğit Şık

Contact: yigit6958@gmail.com

Git: https://github.com/YigitSIK/Comp2Dust

2.Elevator Pitch

Comp2Dust online image optimizer uses various image compression engines to optimize and modify JPEG, PNG and GIF files while keeping the required level of quality.

3.Overview

JPEG, GIF and PNG formats make most of the entire internet's image traffic. That is why this web site will be aiming these three format. Due to increasing upload numbers each day, it become a need to optimize content.

Comp2Dust web site is an online image compressor available for every user who has an internet connection and any popular browser such as Firefox, Mozilla, Chrome ... Our intended users are people who need smaller sizes for their environment. And since we have covered three main formats that today's internet consist of, we have common users.

4.Requirements

- MongoDB database with Atlas cloud service to keep image data but not images themselves.
- In terms of user interface, HTML, CSS, and Bootstrap library will be used to illustrate the system attractively.
- Server side compression function depends on npm package called “compress-images”.
- Another npm package to called “body-parser” to get parsed information from HTML to server side.
- Axios library will be used for sending http requests.
- Multer library for uploading multi-part form data.
- Zip-local npm package to zip folders
- On the server side, express js, and its middlewares, will be used to handle routes and mongoose to connect database.
- Site will be hosted on Heroku servers.

The System’s core functionality is provided by npm package “compress-images”.

The package provides several compression engines as APIs.

Comp2Dust image compressor uses fallowing engines due to their consistency:

JPG Engines:

- mozjpeg
- jpegtran

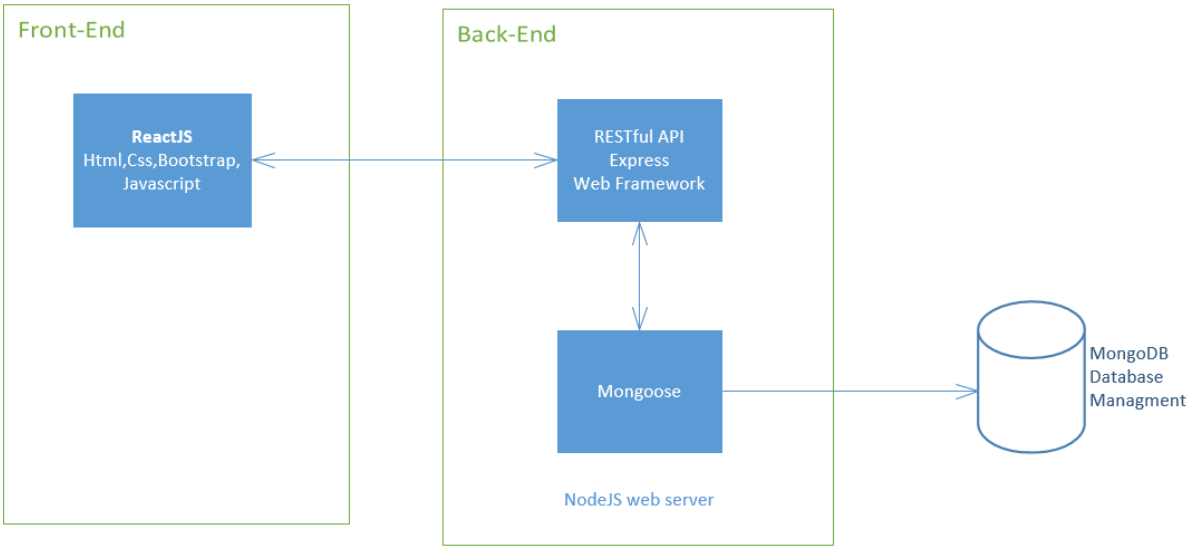
PNG Engines:

- pngquant

GIF Engines:

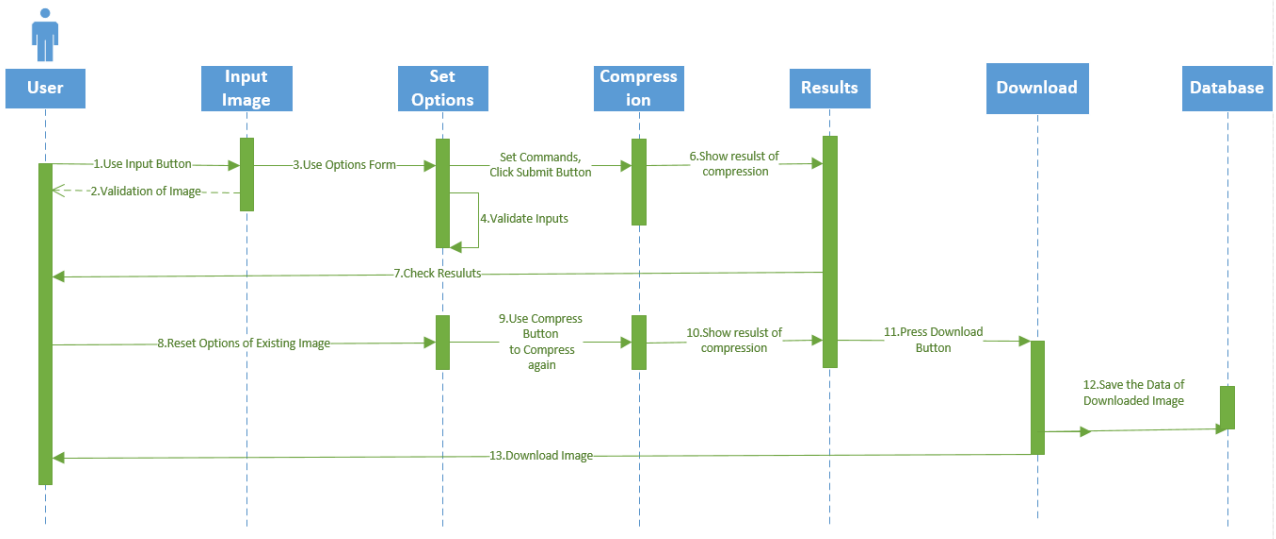
- gifsicle

Block Diagram of the system is given below



5.Functionality

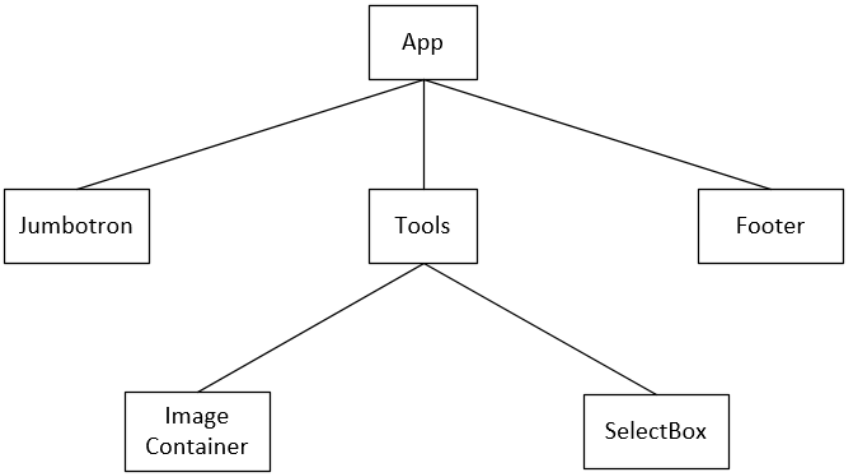
Here is an example to general flow of the application



6.Design

Comp2Dust is a single page web application that has been designed with ReactJS. Thus it is highly modifiable and good looking.

Components tree of Comp2Dust is given below:



App: Covers all content

Jumbotron: Contains Header part of the application

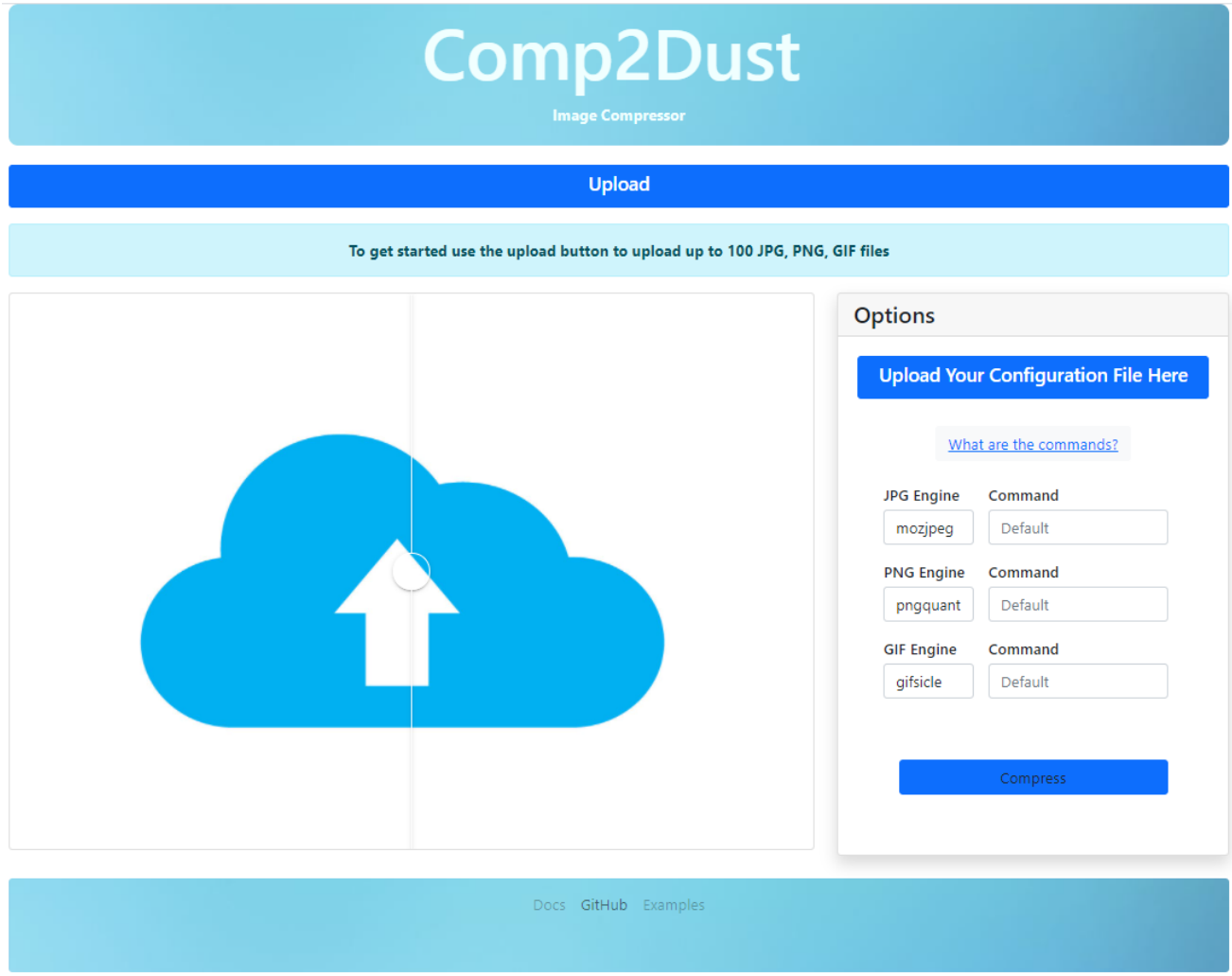
Tools: This is where magic happens, contains all the image displayers and options

Footer: This is the bottom part of the application, contains nav-bar links

Image Container: The Main image displayer with comparison feature is here as an image gallery.

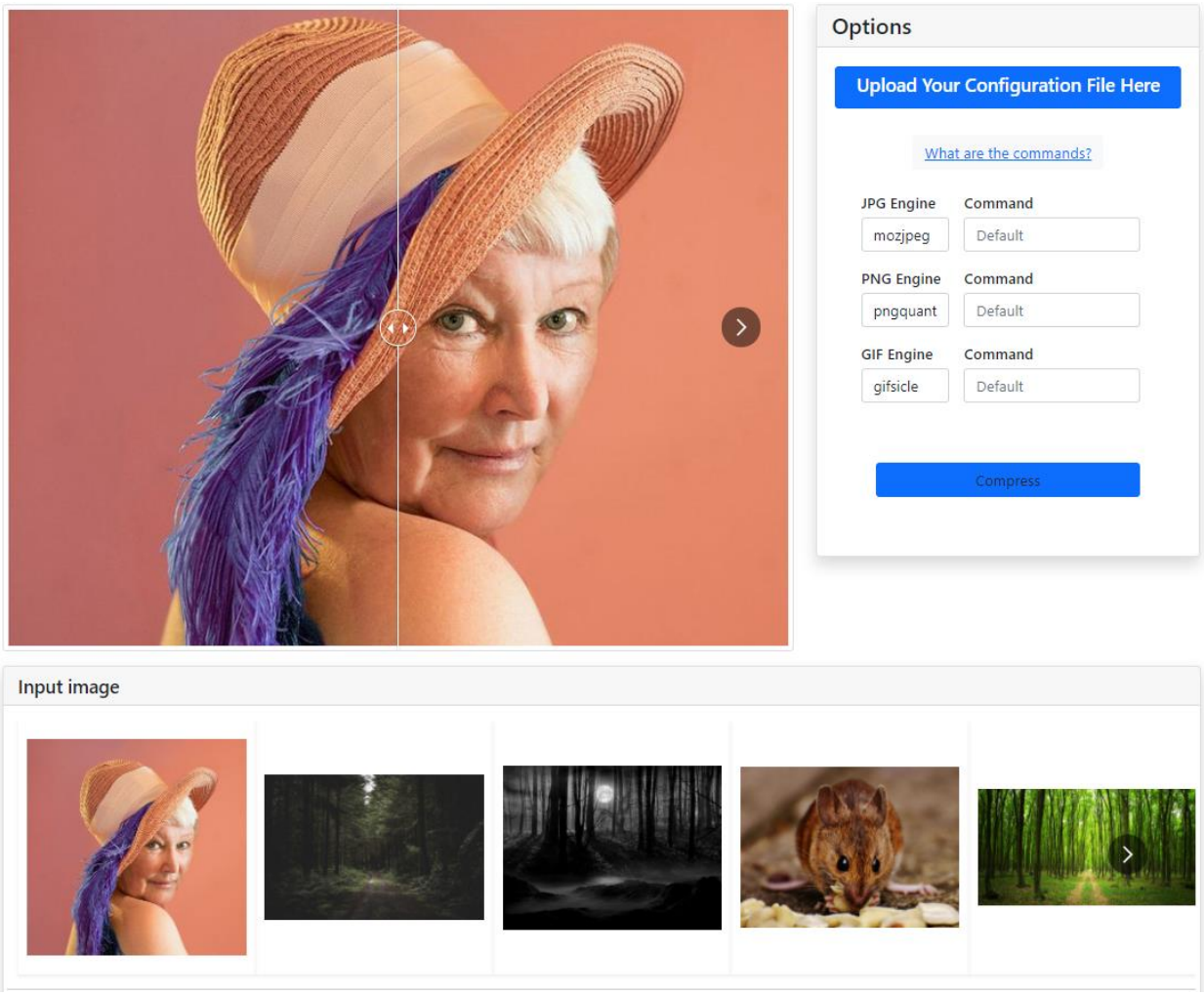
SelectBox: At this part of the application user enters commands, chooses compression engines.

At first glance to app, there is a pretty landing page like the one below

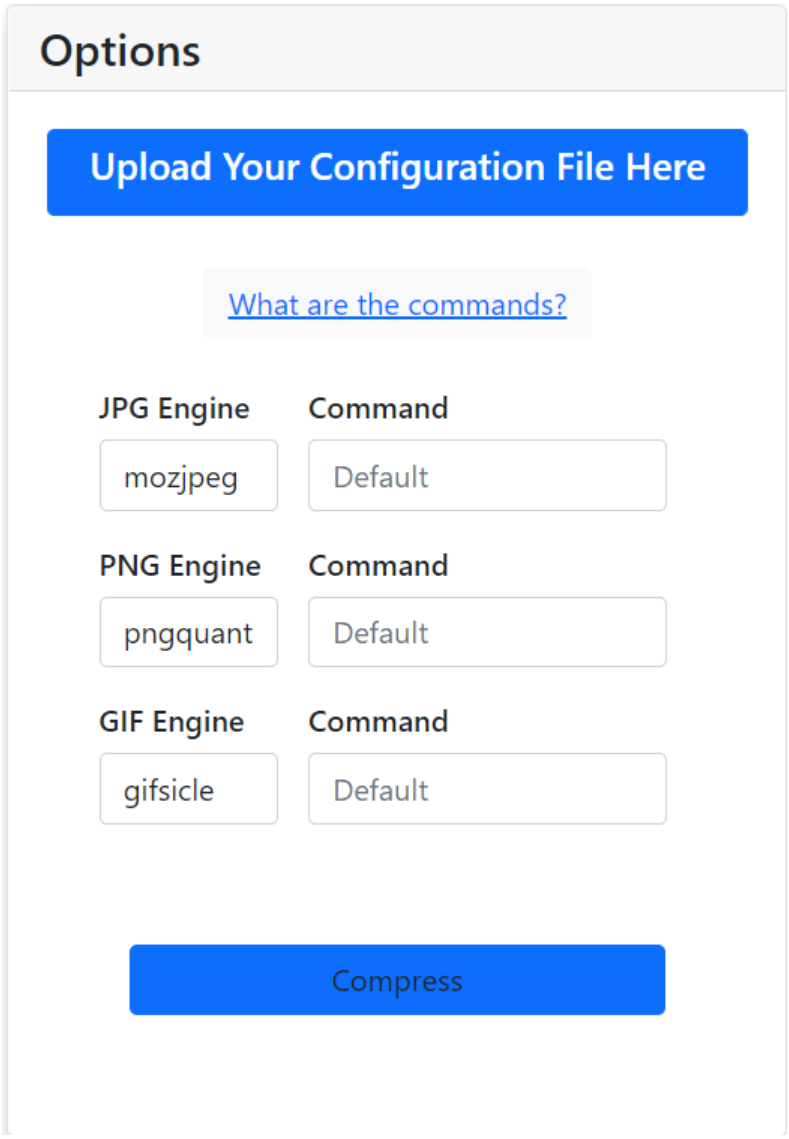


As the Info Box says, user starts with uploading his images through the input box.

Right after user see upload progress through progress bar which will show up below. On images loaded, another carousel displays those images

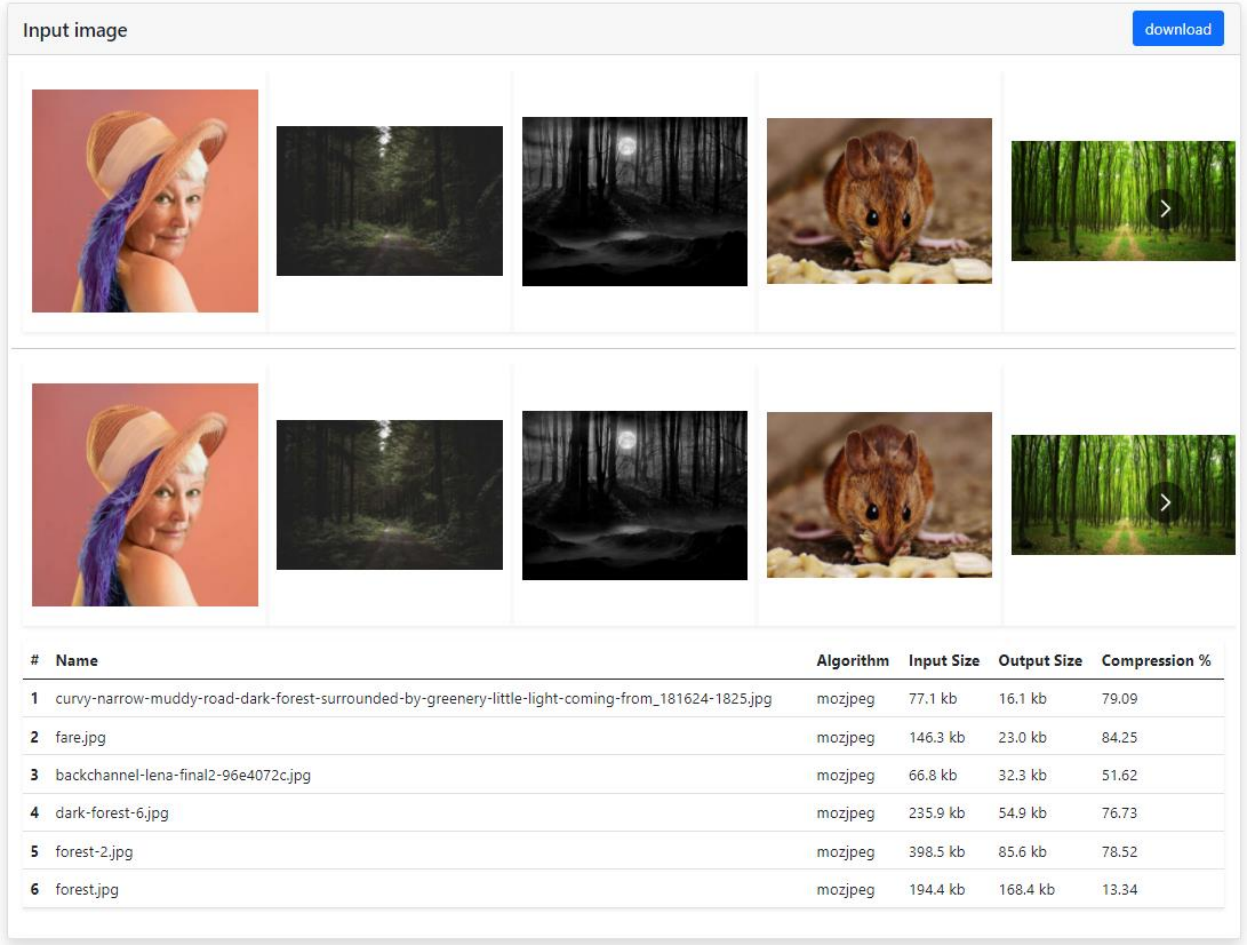


Application provides a graphical user interface that takes user inputs, transferring them to compression API.



- Commands of the application will be discussed in user manual

After compression, results are displayed with statistics and ready to download.

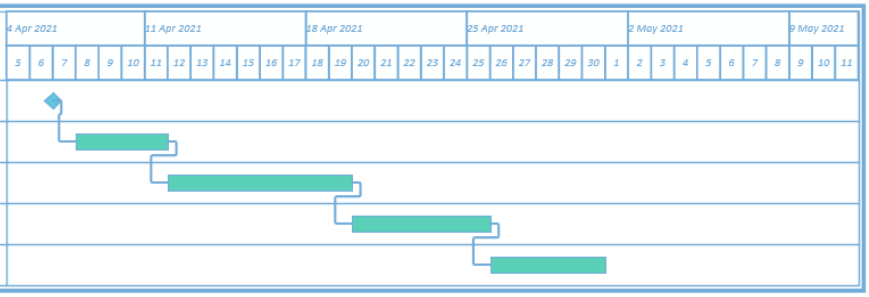


#	Name	Algorithm	Input Size	Output Size	Compression %
1	curvy-narrow-muddy-road-dark-forest-surrounded-by-greenery-little-light-coming-from_181624-1825.jpg	mozjpeg	77.1 kb	16.1 kb	79.09
2	fare.jpg	mozjpeg	146.3 kb	23.0 kb	84.25
3	backchannel-lena-final2-96e4072c.jpg	mozjpeg	66.8 kb	32.3 kb	51.62
4	dark-forest-6.jpg	mozjpeg	235.9 kb	54.9 kb	76.73
5	forest-2.jpg	mozjpeg	398.5 kb	85.6 kb	78.52
6	forest.jpg	mozjpeg	194.4 kb	168.4 kb	13.34

7.Milestones

Alpha

Id	Task Name	Starts	Ends	Time
1	Landing Page Creation	4/7/2021	4/7/2021	0g
2	Basic Implementation	4/8/2021	4/11/2021	4g
3	Binding Database	4/12/2021	4/19/2021	8g
4	General Improvement	4/20/2021	4/25/2021	6g
5	Alpha	4/26/2021	4/30/2021	5g



- As a first task there will be the basic design of constructed landing page.
- By the end of 2nd task application should perform the main function crudely
- At third step working algorithm should be able to send data to MongoDB database.
- At fourt step, any occurence of bugs should be fixed, input validations and exception handling task should be implemented.
- In Alpha section application should be properly working and its user interface should be revised for presentation.

Final

Id	Task Name	Starts	Ends	Time
1	Recreate front-end with React	5/11/2021	5/18/2021	8g
2	Set Basic Upload Feature	5/19/2021	5/19/2021	1g
3	Implement Core Function	5/20/2021	5/20/2021	1g
4	Set-Up Displayers	5/21/2021	5/21/2021	1g
5	Set Download Feature	5/22/2021	5/22/2021	1g
6	Deployment to Heroku	5/23/2021	5/23/2021	1g
7	Set Session Feature	5/24/2021	5/25/2021	2g
8	Improve UI	5/26/2021	5/26/2021	1g
9	Final Version	5/27/2021	5/27/2021	1g