# Student ID: IT21479250

Student Name: Gunathilaka I.U.

# DS – Assignment II

**Assignment Title: Development of a React Frontend Application Using NASA APIs**

**Context**

1. Chosen APIs
2. Challenges faced
3. How the challenges solve

**Chosen APIs**

1. DONKI

The [Space Weather Database Of Notifications, Knowledge, Information (DONKI)](https://ccmc.gsfc.nasa.gov/tools/DONKI/) is a comprehensive on-line tool for space weather forecasters, scientists, and the general space science community. DONKI chronicles the daily interpretations of space weather observations, analysis, models, forecasts, and notifications provided by the Space Weather Research Center (SWRC), comprehensive knowledge-base search functionality to support anomaly resolution and space science research, intelligent linkages, relationships, cause-and-effects between space weather activities and comprehensive webservice API access to information stored in DONKI.

Notifications:

**parameters:**  
'startDate' and 'endDate' are in format 'yyyy-MM-dd' UT  
'type' could be: all, FLR, SEP, CME, IPS, MPC, GST, RBE, report

**Note:**  
'startDate' if left out would default to 7 days prior to the current UT date  
'endDate' if left out would default to current UT date  
'type' if left out would default to 'all'  
The request date range is limit to 30 days. If the request range is greater than 30 days, it would limit your request range to (endDate-30) to endDate.

**Example:**  
<https://api.nasa.gov/DONKI/notifications?startDate=2014-05-01&endDate=2014-05-08&type=all&api_key=DEMO_KEY>

1. NASA Image and Video Library

Use this API to access the NASA Image and Video Library site at [images.nasa.gov](https://images.nasa.gov/" \l "/). For the latest documentation, please go [here](https://images.nasa.gov/docs/images.nasa.gov_api_docs.pdf).

The images.nasa.gov API is organized around REST, has predictable/resource­-oriented URLs and uses HTTP response codes to indicate API errors. This API uses built-­in HTTP features such as HTTP authentication and HTTP verbs, which are understood by many off-­the-­shelf HTTP clients. Please note that JSON is returned by all API responses, including errors. Each of the endpoints described below also contains example snippets which use the curl command­-line tool, Unix pipelines, and the python command­-line tool to output API responses in an easy­ to ­read format.

**Available Endpoints**

The images API contains 4 endpoints GET https://images-api.nasa.gov

A table of text with black text

Description automatically generated with medium confidence

For complete usage information and detailed examples, please visit the [NASA Image and Video Library API documentation](https://images.nasa.gov/docs/images.nasa.gov_api_docs.pdf).

**Challenges faced**

1. What are the technologies that working with the React? Because of there are lot of technologies that working with Reach, I have to spent considarable time to find what is the best match technology to the senario.
2. What kind of database that I need to use. Sql or non-sql?
3. Find a free solution for the application host

**How the challenges solve**

1. I found number of technologies that frequently use in the market like Redux, NextJs, and so on. From there, I picked the NextJs which is one of framework of the React. With that we can use TailwindCSS as well.
2. When DB selection, in the assignment documentation mentioned that need to focus Frontend rather than the Backend, I picked MongoDB for user management. With MongoDB we can handle large number of entries as a documents and we can expand the application anytime.
3. When trying to find a solution for this React base application I have to tryout number of platforms. Finally, I picked the Digital Ocean. There is a student package, to get that we must connect student package active GitHub account. That Digital Ocean package offered; free $200 credits valid for 1 year.