SKY SCRAPING

A MINI-PROJECT REPORT

Submitted by:

Akshita Mishra - ENG17CS0023
Alfiya Anjum M - ENG17CS0025
Amit Kumar Gupta - ENG17CS0026
Amulya J Yadav - ENG17CS0031

of

BACHELOR OF TECHNOLOGY

in

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

at

DAYANANDA SAGAR UNIVERSITY

SCHOOL OF ENGINEERING, BANGALORE-560068

7th SEMESTER

(Course Code: 16CS471)

WEB PROGRAMMING LAB

DAYANANDA SAGAR UNIVERSITY

CERTIFICATE

This is to certify that the Web Programming Mini-Project report entitled ***"SKY SCRAPING"*** *being submitted by* ***Akshita Mishra, Alfiya Anjum M, Amit Kumar Gupta and Amulya J Yadav*** *to Department of Computer Science and Engineering, School of Engineering, Dayananda Sagar University, Bangalore, for the 7th semester B.Tech C.S.E of this university during the academic year*

2020-2021.

Date:_____ _____

Signature of the Faculty in Charge

_____

Signature of the Chairman

ACKNOWLEDGEMENT

We are pleased to acknowledge **Prof. Gousia Thahniyath, Assistant Professor,** Department of Computer Science & Engineering for his invaluable guidance, support, motivation and patience during the course of this mini- project work.

We extend our sincere thanks to **Dr. Sanjay Chitnis** , **Chairman,** Department of Computer Science & Engineering who continuously helped us throughout the project and without his guidance, this project would have been an uphill task.

We have received a great deal of guidance and co-operation from our friends and we wish to thank one and all that have directly or indirectly helped us in the successful completion of this mini-project work.

Team Members

Akshita Mishra	**ENG17CS0023**
Alfiya Anjum M	**ENG17CS0025**
Amit Kumar Gupta	**ENG17CS0026**
Amulya J Yadav	**ENG17CS0031**

Abstract

A Sky-Scraper is a JavaScript-based project which is data scraping used for extracting data from NASA websites. The goal of the project is to fetch the latest images and videos along with their information from NASA's websites.Many things become clearer when seen from above, and Earth is no exception. Images of Earth from space provide information that cannot be obtained any other way, and these images continue to make important contributions to science and commerce.

The space station's unique orbit offers views that differ from those of traditional Earth-viewing satellites. One of the most popular websites at NASA is the Astronomy Picture of the Day. In fact, this website is one of the most popular websites across all federal agencies.This endpoint structures the APOD imagery and associated metadata so that it can be repurposed for other applications. In addition, if the concept_tags parameter is set to True, then keywords derived from the image explanation are returned. These keywords could be used as auto-generated hashtags for twitter or instagram feeds; but generally help with discoverability of relevant imagery.

An ever expanding diversity and availability of remote sensing data—from the space station, small satellite constellations, and even drone technology—provide vast, complex data sets, and also drive a need for data processing advances. Although previously, only experts in the field performed capture and analysis of Earth images from space, big data has made its way into the hands of the larger community. Making this wealth of information useful requires rapid innovation in computing technology.

TABLE OF CONTENTS

CHAPTER 1

1. INTRODUCTION

1.1 PROBLEM STATEMENT

The aim is to build automated 'Sky Scraping'. The purpose is to develop a website to fetch the latest astronomy pictures updated by NASA.

1.2 OBJECTIVES OF THE PROJECT

The main objective of the Project on Sky Scraping is to fetch the details of different astronomy and space science related images featured each day, along with a brief explanation. Each day, a different image or photograph of our fascinating universe is featured, along with a brief explanation written by a professional astronomer.

CHAPTER 2

2. SYSTEM REQUIREMENTS

2.1 FUNCTION REQUIREMENTS

Provides the searching facilities for latest images and videos along with their information from NASA's websites.The website should display all the images and related information of the past 10 days. It tracks all the information of the posted images and manages the information. Shows the information and description of what NASA has posted about it.

2.2 SOFTWARE REQUIREMENTS

- Text Editor(Visual Studio Code)
- Google Chrome (Version 80+)

2.3 HARDWARE REQUIREMENTS

- Processor : 64bit
- RAM : 2GB
- Hard Disk : 256GB

CHAPTER 3

3. SYSTEM DESIGN

3.1 ARCHITECTURE / DATA FLOW DIAGRAMS

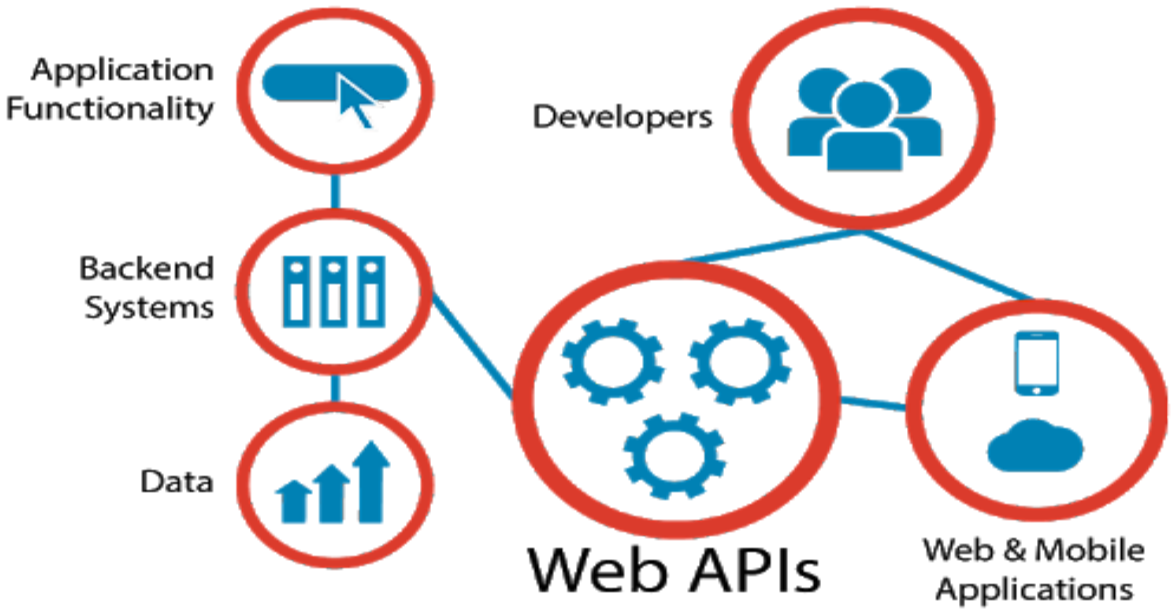

Fig.3.1. How APIs work?

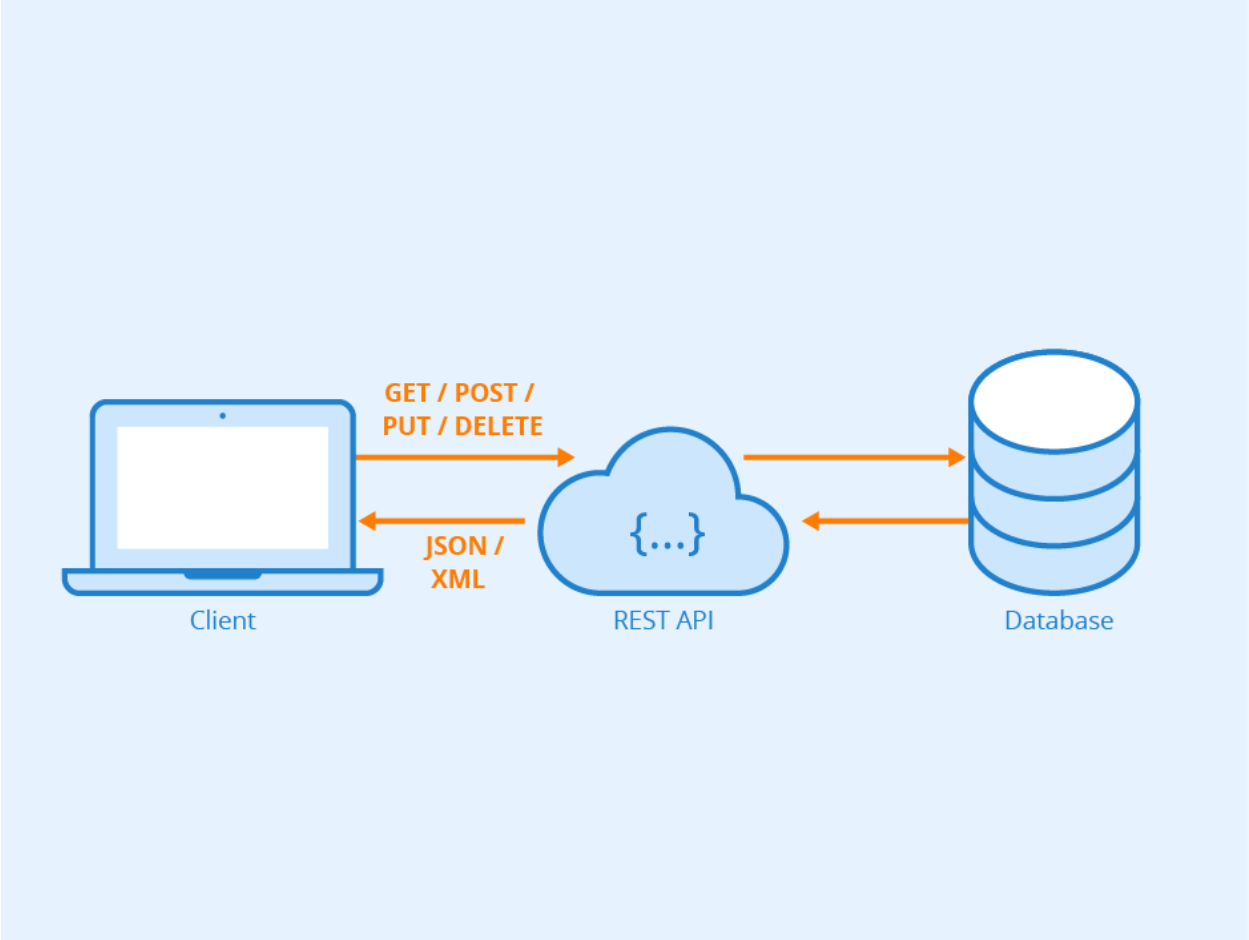

Fig.3.2. Implementation of NASA API

3.2 MODULES

JavaScript Functions Used:

getFullYear()

getMonth()

getDate()

XMLHttpRequest()

Http.open("GET", url);

Http.send();

addEventListener()

CHAPTER 4

4. SYSTEM IMPLEMENTATION

4.1 MODULE DESCRIPTION

getFullYear()

The getFullYear() method returns the year (four digits for dates between year 1000 and 9999) of the specified date.

getMonth()

The getMonth() method returns the month (from 0 to 11) for the specified date, according to local time.

getDate()

The getDate() method returns the day of the month (from 1 to 31) for the specified date.

XMLHttpRequest()

The XMLHttpRequest object can be used to request data from a web server.

Http.open("GET", url);

The XMLHttpRequest method open() initializes a newly-created request, or re-initializes an existing one. The HTTP request method to use, such as "GET", "POST", "PUT", "DELETE", etc. Ignored for non-HTTP(S) URLs.

url - A DOMString representing the URL to send the request to.

Http.send();

The XMLHttpRequest method send() sends the request to the server. If the request is asynchronous (which is the default), this method returns as soon as the request is sent and the result is delivered using events. If the request is synchronous, this method doesn't return until the response has arrived.

send() accepts an optional parameter which lets you specify the request's body; this is primarily used for requests such as PUT. If the request method is GET or HEAD, the body parameter is ignored and the request body is set to null.

addEventListener()

The addEventListener() method attaches an event handler to the specified element. When using the addEventListener() method, the JavaScript is separated from the HTML markup, for better readability and allows you to add event listeners even when you do not control the HTML markup.

4.2 PSEUDOCODE

INDEX.HTML

```
<!doctype html>

<html lang="en">



<head>

  <meta charset="utf-8">

  <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">

  <!-- Google fonts -->

  <link
href="https://fonts.googleapis.com/css?family=Audiowide|Montserrat&display
=swap" rel="stylesheet">

  <!-- My CSS -->

  <link rel="stylesheet" href="styles.css">

  <title>Images from Space</title>

  <link rel="icon" href="faviconN.ico">

</head>



<body>

  <header class="header">

    <h1 class="heading-primary">Images from Space</h1>

    <h3 class="heading-secondary">Data retrieved from NASA's APOD API</h3>

  </header>

  <div class="container">
```

```
    <div class="item"><a href="#popup-img" class="popup-link"
id="popup-link-10"><img src="placeholder.png" class="img" id="img--10"
alt="image from NASA API"></a></div>

    <div class="item"><a href="#popup-img" class="popup-link"
id="popup-link-9"><img src="placeholder.png" class="img" id="img--9"
alt="image from NASA API"></a></div>

    <div class="item"><a href="#popup-img" class="popup-link"
id="popup-link-8"><img src="placeholder.png" class="img" id="img--8"
alt="image from NASA API"></a></div>

    <div class="item"><a href="#popup-img" class="popup-link"
id="popup-link-7"><img src="placeholder.png" class="img" id="img--7"
alt="image from NASA API"></a></div>

    <div class="item"><a href="#popup-img" class="popup-link"
id="popup-link-6"><img src="placeholder.png" class="img" id="img--6"
alt="image from NASA API"></a></div>

    <div class="item"><a href="#popup-img" class="popup-link"
id="popup-link-5"><img src="placeholder.png" class="img" id="img--5"
alt="image from NASA API"></a></div>

    <div class="item"><a href="#popup-img" class="popup-link"
id="popup-link-4"><img src="placeholder.png" class="img" id="img--4"
alt="image from NASA API"></a></div>

    <div class="item"><a href="#popup-img" class="popup-link"
id="popup-link-3"><img src="placeholder.png" class="img" id="img--3"
alt="image from NASA API"></a></div>

    <div class="item"><a href="#popup-img" class="popup-link"
id="popup-link-2"><img src="placeholder.png" class="img" id="img--2"
alt="image from NASA API"></a></div>

    <div class="item"><a href="#popup-img" class="popup-link"
id="popup-link-1"><img src="placeholder.png" class="img" id="img--1"
alt="image from NASA API"></a></div>

  </div>
```

```
  <footer>
    <p id="copyright"></p>
    <script>
      document.getElementById("copyright").innerHTML = "© Copyright Freeda Moore " + new Date().getFullYear();
    </script>
  </footer>

  <!-- popup -->
  <div class="popup" id="popup-img">
    <div class="popup-content">
      <div class="popup-item">
        <img src="" id="pop-img" alt="image from NASA API">
        <iframe id="pop-vid" width="100%" height="100%" src=""></iframe>
        <p id="img-copyright"></p>
      </div>
      <div class="popup-item popup-text">
      <h3 id="title">Title</h3>
      <h4 id="date">Date</h4>
      <p id="explanation">Explanation</p>
      <a href="#" class="popup-close">Close</a>
      </div>
    </div>
  </div>
```

```
  <script src="script.js"></script>
</body>


</html>
```

SCRIPTS.JS

```
const sevenDaysAgo = new Date((new Date()).valueOf() - 1000 * 60 * 60 * 24 * 10);

const year = sevenDaysAgo.getFullYear();

const month = sevenDaysAgo.getMonth() + 1;

const day = sevenDaysAgo.getDate();

const startDate = `${year}-${month}-${day}`;

const apiKey = 'nd2RPjmuvvnuwJQSf6fKoOW7T9jGuXUWzUv16oNX';

const url = `https://api.nasa.gov/planetary/apod?start_date=${startDate}&api_key=${apiKey}`;

// HTTP request

const Http = new XMLHttpRequest();

Http.open("GET", url);

Http.send();


Http.onreadystatechange = function() {
```

```
if (this.readyState == 4 && this.status == 200) {
  const dataArray = JSON.parse(Http.responseText);
  // console.log(dataArray);
  //iterate through json items and populate images on page
  let i;
  const length = 10; // we only want 10 images
  for (i = 0; i < length; i++) {

    if (dataArray[i]['media_type'] === "video") {
      document.getElementById(`img--${i+1}`).src = "video-img.png";
    } else {
      document.getElementById(`img--${i+1}`).src = dataArray[i].url;
    }
  }

  //display popup elements
  const images = document.getElementsByClassName('popup-link');
  const popImg = document.getElementById('pop-img');
  const popVid = document.getElementById('pop-vid');
  const imgCopyright = document.getElementById('img-copyright');
  const title = document.getElementById('title');
  const date = document.getElementById('date');
  const explanation = document.getElementById('explanation');
```

```
    for (let i = 0; i < length; i++) {
      images[i].addEventListener("click", function() {
        //images are displayed in reverse date order, so need to make up
for this
        let x = length-i-1;
        if (dataArray[x]['media_type'] === "video") {
          popImg.style.display = "none";
          popVid.style.display = "block";
          popVid.src = dataArray[x].url;
        } else {
          popImg.style.display = "block";
          popVid.style.display = "none";
          popImg.src = dataArray[x].url;
        }
        imgCopyright.textContent = "Image Credit & Copyright: " +
dataArray[x].copyright;
        title.textContent = dataArray[x].title;
        date.textContent = dataArray[x].date;
        explanation.textContent = dataArray[x].explanation;
      })
    }
  }
}
```

CHAPTER 5

OUTPUT SCREEN SHOTS

Fig.5.1 Homepage

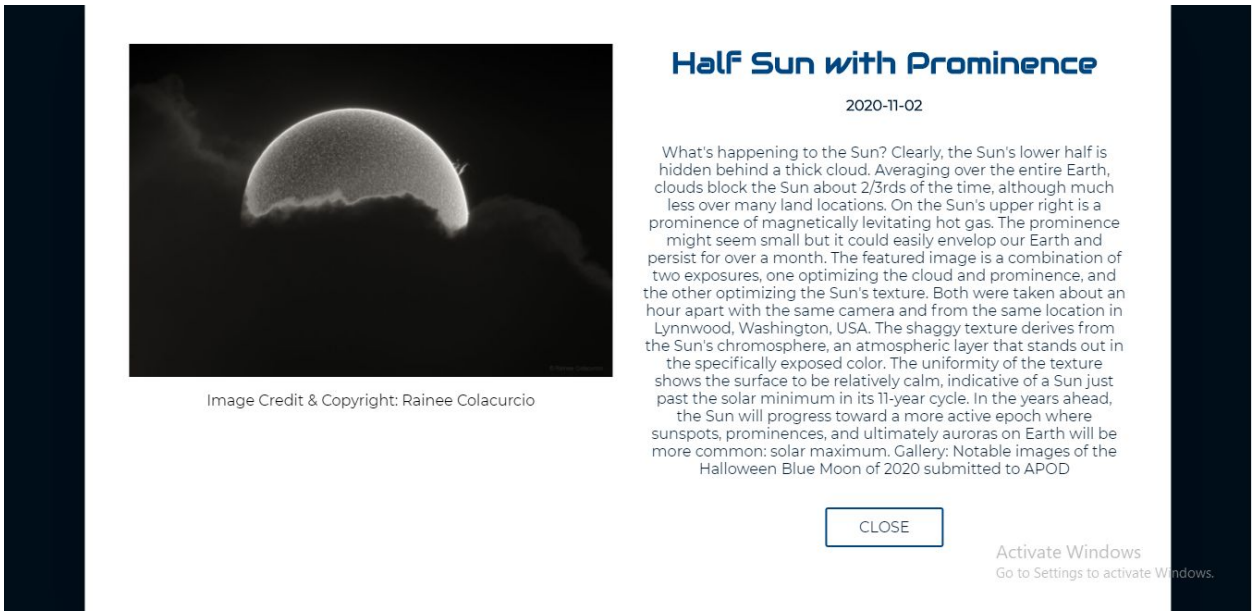

Fig.5.2 A Image with its description

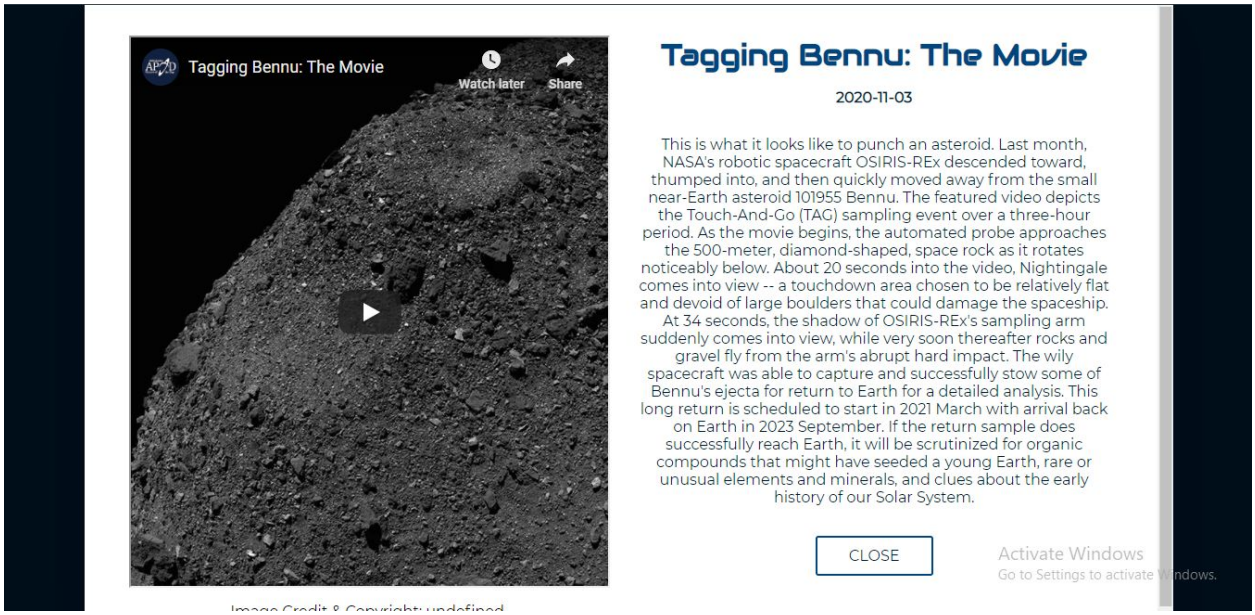

Fig.5.3 A Video with its description

CHAPTER 6

CONCLUSION

Astronomy Picture of the Day (APOD) is a website provided by NASA and Michigan Technological University (MTU). According to the website, "Each day a different image or photograph of our universe is featured, along with a brief explanation written by a professional astronomer." The photograph does not necessarily correspond to a celestial event on the exact day that it is displayed, and images are sometimes repeated. However, the pictures and descriptions often relate to current events in astronomy and space exploration. The text has several hyperlinks to more pictures and websites for more information. The images are either visible spectrum photographs, images taken at non-visible wavelengths and displayed in false color, video footage, animations, artist's conceptions, or micrographs that relate to space or cosmology. Past images are stored in the APOD Archive, with the first image appearing on June 16, 1995. This initiative has received support from NASA, the National Science Foundation, and MTU. The images are sometimes authored by people or organizations outside NASA, and therefore APOD images are often copyrighted, unlike many other NASA image galleries.So the website displays the image from the NASA image galleries with the information provided about the posted image, this helps in collecting and viewing the information about the space.