IMAGE UPLOADER APP

PROJECT REPORT

Full-STACK 2

FullStack Project- II Report
On

IMAGE UPLOADER APP

Department of Computer Engineering & Applications

Institute of Engineering & Technology

Submitted by

Gauri Agrawal	Roll No: 181500238
Nidhi Jain	Roll No: 181500423
Radhika Bansal	Roll No: 181500528
Sakshi Bhardwaj	Roll No: 181500608

Supervised By: -

Mr. Pankaj Kapoor

Department of Computer Engineering & Application

Department of Computer Engineering and Applications

GLA University, 17 km. Stone NH#2, Mathura-Delhi Road,
Chaumuha, Mathura – 281406 U.P (India)

Declaration

we hereby declare that the work which is being presented in the B.Tech. Project **“PHOTO GALLERY APP”**, in partial fulfilment of the requirements for the award ofthe ***Bachelor of Technology*** in Computer Science and Engineering and submitted to the Department of Computer Engineering and Applications of GLA University, Mathura, is an authentic record of my/our own work carried under the supervision of **Mr. Pankaj Kapoor.**

The contents of this project report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree.

Name of Candidate: Gauri Agrawal
University Roll No.: 181500238

Name of Candidate: Radhika Bansal
University Roll No.: 181500528

Name of Candidate: Nidhi Jain
University Roll No.: 181500423

Name of Candidate: Sakshi Bhardwaj
University Roll No.:181500608

Acknowledgement

In the present world of competition there is a race of existence in which those are having will to come forward succeed. Project is like a bridge between theoretical and practical working. With this willing we joined this particular project in our study. It gives us great pleasure and we are glad to represent the project report of our Full-stack-2 Project undertaken during the third year of our graduation era.

It has indeed been a great privilege for us to have Mr. Pankaj Kapoor Department of Computer Engineering & Application, Institute of Engineering & Technology, GLA University, Mathura, as our mentor in this Project. His awe-inspiring personality, superb guidance and constant encouragement are the motive force behind this project. We take this opportunity to express our utmost gratitude to him. We are also indebted to him for his timely and valuable advise.

We are highly grateful to Prof. Anand Singh Jalal, Head, Department of Computer Engineering & Application for providing necessary facilities and encouraging us during the course of work. We are thankful to all technical and non-technical staff of the Department of the Computer Science and Engineering for their constant assistance and their co-operation.

Index

Requirements

Hardware Requirements

- Personal computer
- Laptop
- Android Phone

Note: Any one of these devices is required to run our Project.

Software Requirements

Theapplication software which we required to build our website are as following:

- ***VsCode (Visual Code Studio)***
- ***MongoDB***

Project Introduction:

We are creating an image uploading and viewing application in which user can:

- Upload an image of one’s own choice and displaying the preview of the image before uploading.
- Getting the notification of successfully uploading the image.
- Viewing all the uploaded images in the gallery section.

We will be creating a soothing, eye-captivating and simple user interface using react which will indulge the user and provides a platform to showcase their photography skills. This project will be a unembellished one, we will not be adding much functionalities for now but will surely have a good scope for expanding it into a major one.

Problem Statement:

For any developer who envisions building an application, uploading images is a major component they have to take into account. It is an essential requirement while creating a complete application. File uploading means a user from a client machine wants to upload files to the server. For example, users can upload images, videos, etc on Facebook, Instagram. As with any programming problem, there are many ways to achieve this outcome. That’s why we are creating such application on a smaller scale. We will use several technologies and techniques to provide the main functionality like NPM as a package manager, Axios to save file and get files, Multer Middleware for handling multipart/form-data, etc.

Working Methodology:

The resources used for designing and developing this project are the back-end techniques which have to be implemented on a text editor. The process of uploading an image can be broadly divided into two steps:

Select a File (user input): To enable the user to pick a file, the first step is to add the tag to our App component. This tag should have the type attribute set as “file”. Now, we need an event handler to listen to any changes made to the file. This event handler will be triggered whenever the user selects a new file and will update the state.

Send a request to the server: After storing the selected file (in the state), we are now required to send it to a server. For this purpose, we can use fetch or Axios. (In this code, we use Axios a promisebased HTTP client for the browser and NodeJS). The file is sent to the service wrapped in a FormData object.

Firstly, we prepare a design format for the application of how our website will look like and what functionality will cover which area of the website. Then, we will work on how all these modules get connected and provide a better interface for interactivity and looks astonishing at the very first sight of the user. We have been using MERN (MongoDB, ExpressJS, Reactjs and Nodejs) Stack in this project.

The technologies used are:

React: Uploader Gallery prepares your images for the gallery on a React-based application. React.js is now one of the most battletested and matured frontend frameworks in the world and express.js is it's counterpart among backend/server frameworks. If you're building an app today, you can't pick a better duo than this. React is a declarative, efficient, and flexible JavaScript library for building user interfaces. It lets you compose complex UIs from small and isolated pieces of code called “components”. We are going to design different components for our application where the related content renders.

MongoDB: MongoDB is a general purpose, document-based, distributed database built for modern application developers and for the cloud era. MongoDB is a document database, which means it stores data in JSON-like documents. We believe this is the most natural way to think about data, and is much more expressive and powerful than the traditional row/column model. We will be using it to store the images uploaded by the users. We need to deploy it on the cloud to make it work so we use MongoDB Atlas for this, that stores the images on the cloud or when online.

ExpressJS: Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications. It is flexible as there are numerous modules available on npm, which can be directly plugged into Express. It makes it easier to organize your application’s functionality with middle ware and routing; it adds helpful utilities to Node.js HTTP objects; it facilitates the rendering of dynamic HTTP objects.

Module Description

This project is categorized into two modules:

1. User Module: This module includes a user interface for uploading the images. This will be the main or say front page of the React App. It consists of a form where to get the user input as the images in a specific format which will display the name of the chosen image on the aside box and getting the notification if the user added a suitable picture to be uploaded. It is a client-side module.

2. Gallery Module: This module includes an interface of cards in which the images are displayed that are uploaded by the user. It is the serverside module which takes the images from user and stores them into a database and render them in this module.

Softwares Used:

Visual Studio Code

MongoDB Atlas

Technologies Used

The resources used for designing and developing this project are the back-end techniques which have to be implemented on a text editor.

Vs code

Visual Studio Code is a streamlined code editor with support for development operations like debugging, task running, and version control. It aims to provide just the tools a developer needs for a quick code-build-debug cycle and leaves more complex workflows to fuller featured IDEs, such as Visual Studio IDE.

Visual Studio Code combines the simplicity of a source code editor with powerful developer tooling, like IntelliSense code completion and debugging.

First and foremost, it is an editor that gets out of your way. The delightfully frictionless edit-build-debug cycle means less time fiddling with your environment, and more time executing on your ideas

At its heart, Visual Studio Code features a lightning fast source code editor, perfect for day-to-day use. With support for hundreds of languages, VS Code helps you be instantly productive with syntax highlighting, bracket-matching, auto-indentation, box-selection, snippets, and more. Intuitive keyboard shortcuts, easy customization and community-contributed keyboard shortcut mappings let you navigate your code with ease.

For serious coding, you'll often benefit from tools with more code understanding than just blocks of text. Visual Studio Code includes built-in support for IntelliSense code completion, rich semantic code understanding and navigation, and code refactoring.

And when the coding gets tough, the tough get debugging. Debugging is often the one feature that developers miss most in a leaner coding experience, so we made it happen. Visual Studio Code includes an interactive debugger, so you can step through source code, inspect variables, view call stacks, and execute commands in the console.

VS Code also integrates with build and scripting tools to perform common tasks making everyday workflows faster. VS Code has support for Git so you can work with source control without leaving the editor including viewing pending changes.

MongoDB

MongoDB is an object-oriented, simple, dynamic, and scalable NoSQL database. It is based on the NoSQL document store model. The data objects are stored as separate documents inside a collection instead of storing the data into the columns and rows of a traditional relational database.

Companies and development teams of all sizes use MongoDB because: The document data model is a powerful way to store and retrieve data that allows developers to move fast. MongoDB's horizontal, scale-out architecture can support huge volumes of both data and traffic.

SQL databases are used to store structured data while NoSQL databases like MongoDB are used to save unstructured data. MongoDB is used to save unstructured data in JSON format. MongoDB does not support advanced analytics and joins like SQL databases support.

MongoDB, unfortunately, does not support transactions. So if you need to update more than one document or collection per user request, don't use MongoDB. It may lead to corrupted data, as there is no ACID guarantee. Rollbacks have to be handled by your application.

Advantages of MongoDB

- Schema less − MongoDB is a document database in which one collection holds different documents. Number of fields, content and size of the document can differ from one document to another.
- Structure of a single object is clear.
- No complex joins.
- Deep query-ability. MongoDB supports dynamic queries on documents using a document-based query language that's nearly as powerful as SQL.
- Tuning.
- Ease of scale-out − MongoDB is easy to scale.
- Conversion/mapping of application objects to database objects not needed.
- Uses internal memory for storing the (windowed) working set, enabling faster access of data.

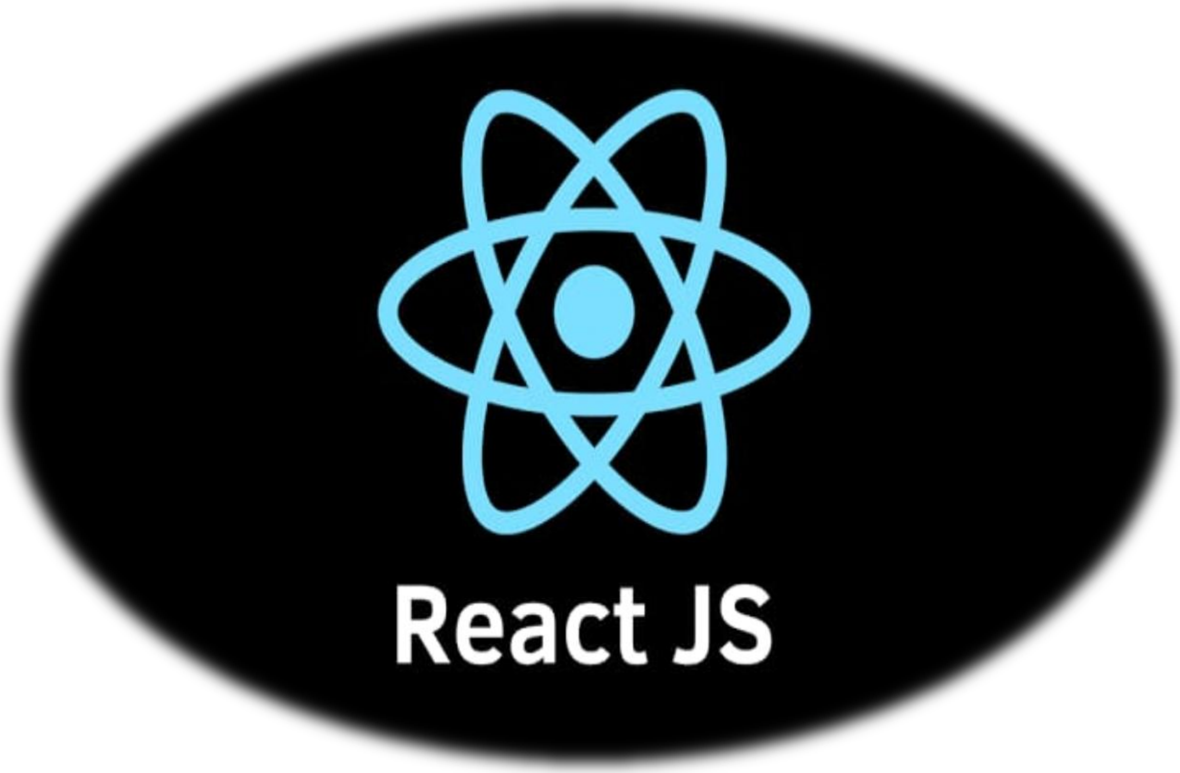
React JS

Reactjs is an open-source JavaScript library that is used for building user interfaces specifically for single-page applications. It's used for handling the view layer for web and mobile apps. **React** also allows us to create reusable UI components.

Is React a Frontend orBackend library?Created and maintained by Facebook, React is a front-end library that runs on a browser. Like most libraries, this one runs on web servers like Apache or with backends like PHP or Rails.

Both HTML and CSS are integral to any web development project. If you have these skills already, then learning React should be a relatively straightforward process. It has its own unique set of challenges, but it is an excellent tool to have in order to start or further your career as a web developer.

Today, ReactJS has become highly popular because of its extra simplicity and flexibility. Many people are even referring to it as the future of web development. It is estimated that more than 1,300 developers and over 94,000 sites utilize ReactJS.



React is a JavaScript library for building user interfaces. React is used to build single page applications. React allows us to create reusable UI components.

Advantages

React allows developers to create large web applications that can change data, without reloading the page. The main purpose of React is to be fast, scalable, and simple. It works only on user interfaces in the application.

Express.JS

Express.js, or simply Express, is a backend web application framework for Node.js, released as free and open-source software under the MIT License. It is designed for building web applications and APIs. It has been called the de facto standard server framework for Node.js.

The express framework is built on top of the node.js framework and helps in fast-tracking development of server-based applications. Routes are used to divert users to different parts of the web applications based on the request made.

ExpressJS **is** a prebuilt NodeJS framework that can help you in creating server-side web applications faster and smarter. Simplicity, minimalism, flexibility, scalability are some of its characteristics and since it **is** made in NodeJS itself, it inherited its performance as well.

Web Applications

Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications.

APIs

With a myriad of HTTP utility methods and middleware at your disposal, creating a robust API is quick and easy.

Performance

Express provides a thin layer of fundamental web application features, without obscuring Node.js features that you know and love.

Frameworks

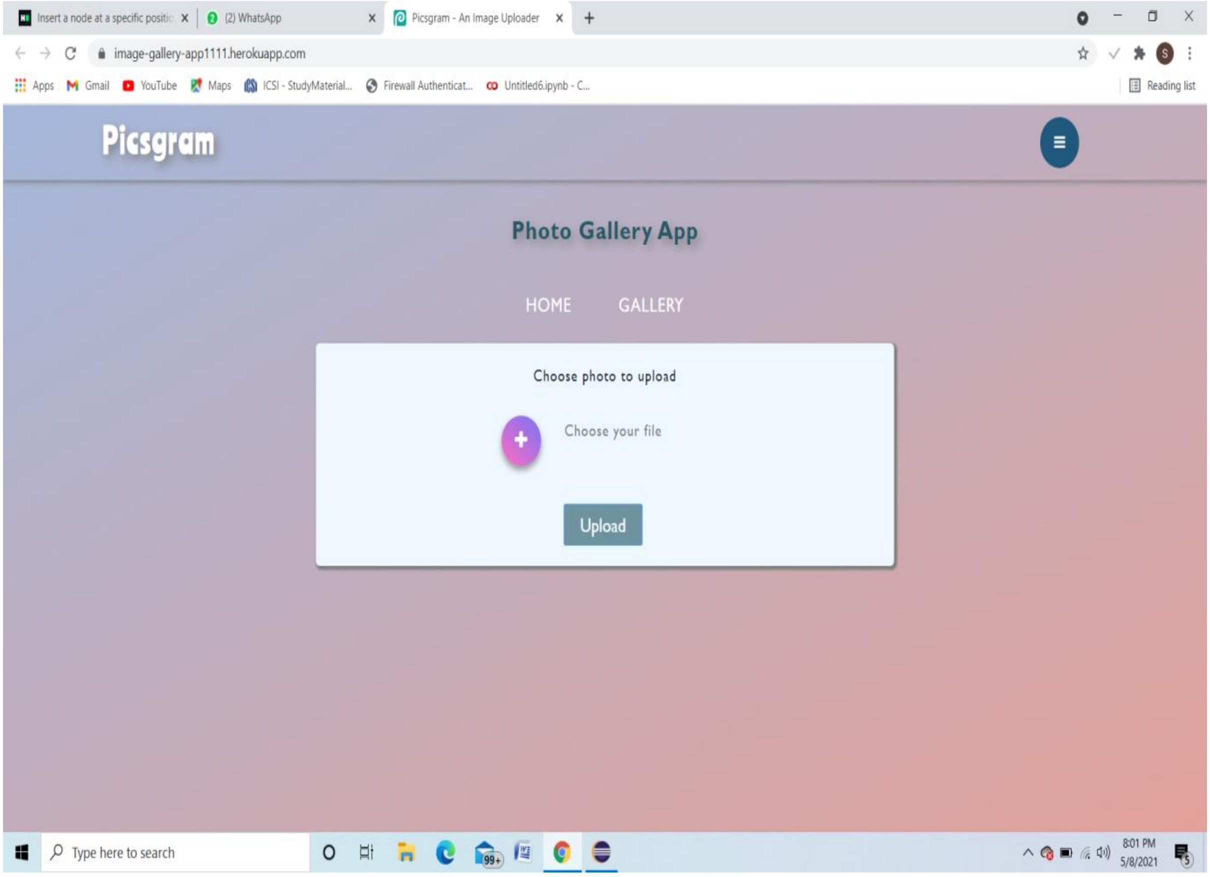
Many popular frameworks are based on Express.

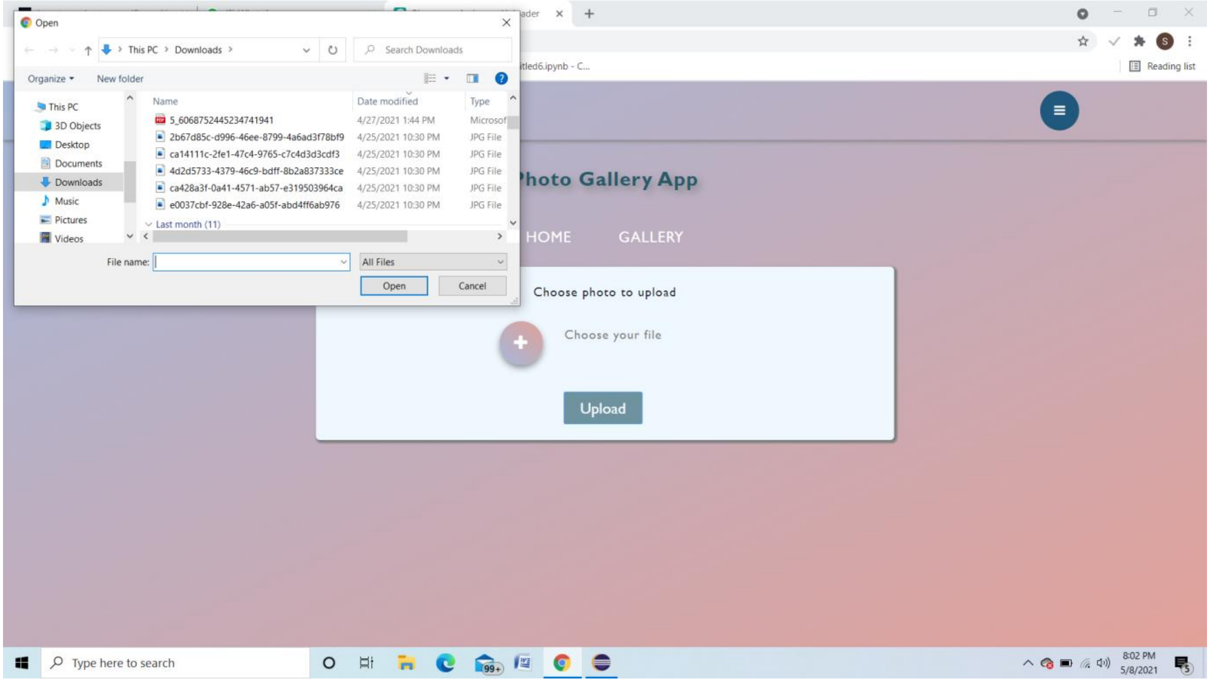
Scope of the Project:

This project is new having a vast interest of people and a necessity in the Web World where people tends to find everything on Internet like variety of images used by people in their projects or making user interface templates and also for editing, this application will be a good option. We can increase the functionality by adding the downloading option to the images so that people can download them in .jpg, .jpeg and .png format and use them according to their requirements like we have seen many such platforms like Pexels, Pixabay, Unsplash, etc. These are the sites accessed in a huge scale. We currently not creating the categories in our project but later on, we can add several categories with respect to the images uploaded or people can upload the images in the particular category option which will make it easy for people to access through the website and get the appropriate results.

Screenshots

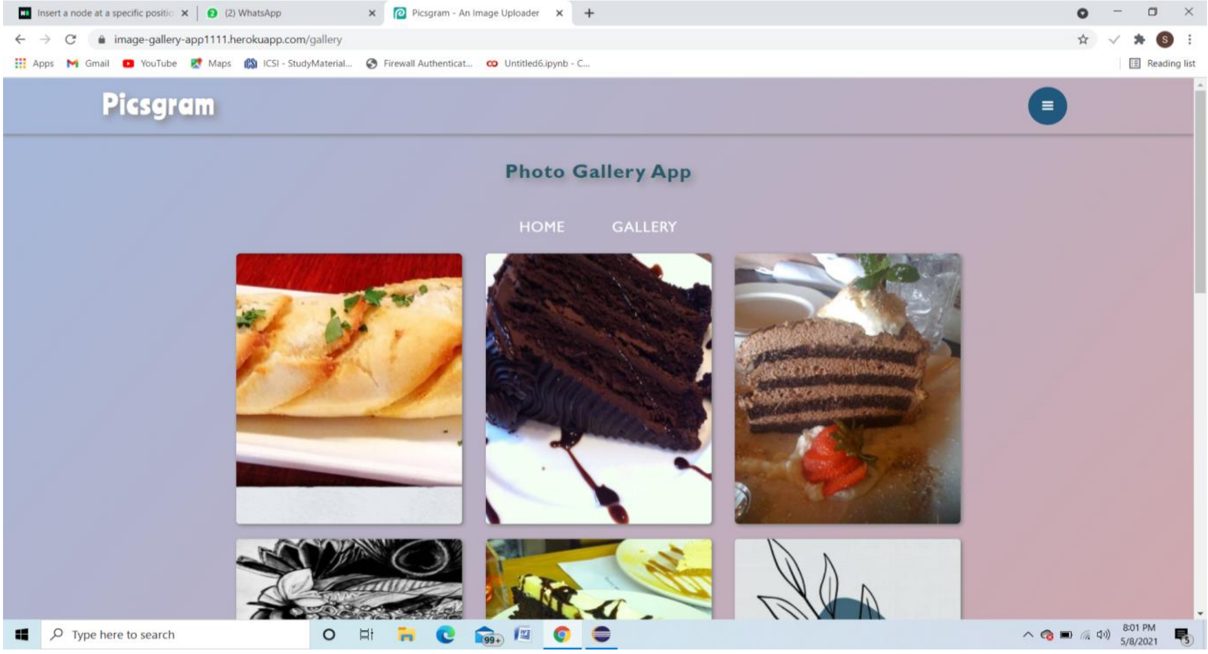
1. Main Page/ Uploading module: This is the front- page of our application where user can upload an image on clicking the plus symbol and also show the name of the file in the sidebar, after which the upload button is enabled and the image will get uploaded with a notification.





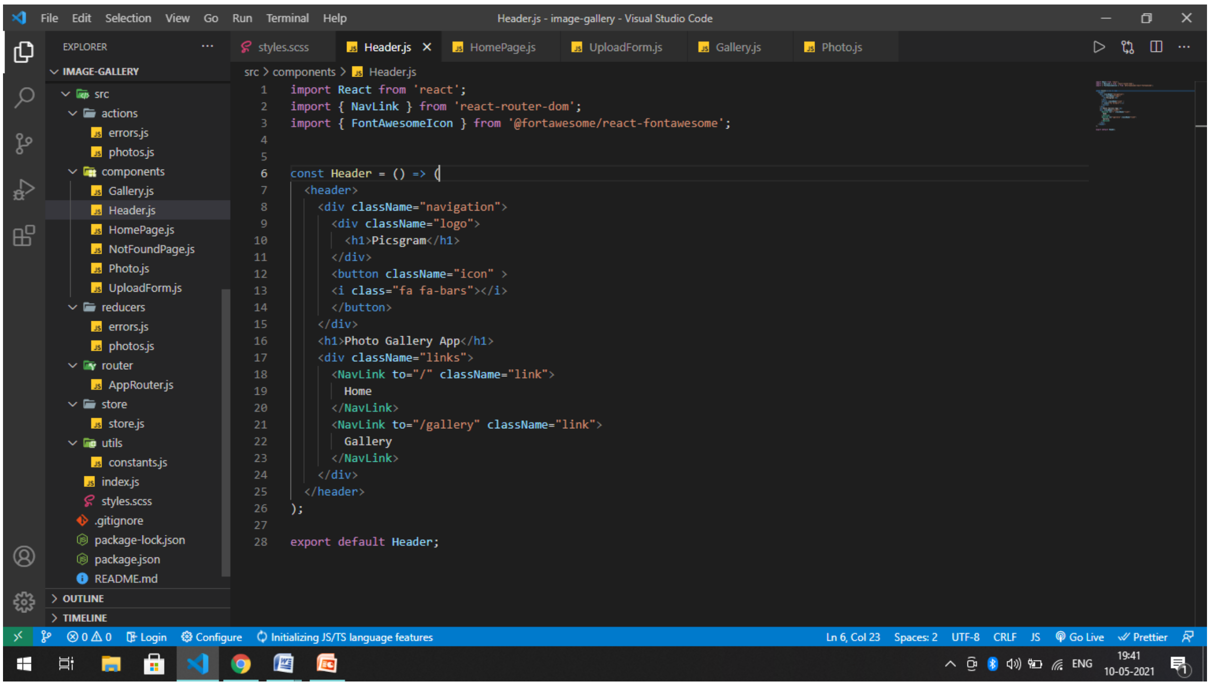
Name	Date modified	Type
5_6068752445234741941	4/27/2021 1:44 PM	Microsof
2b67d85c-d996-46ee-8799-4a6ad3f78bf9	4/25/2021 10:30 PM	JPG File
ca14111c-2fe1-47c4-9765-c7c4d3d3cdf3	4/25/2021 10:30 PM	JPG File
4d2d5733-4379-46c9-bdff-8b2a837333ce	4/25/2021 10:30 PM	JPG File
ca428a3f-0a41-4571-ab57-e319503964ca	4/25/2021 10:30 PM	JPG File
e0037cbf-928e-42a6-a05f-abd4ff6ab976	4/25/2021 10:30 PM	JPG File

2. Gallery Page: This is the Gallery page where the user can see the uploaded image in the form of cards. Firstly, these images are stored in the database and then after rendered on the screen for viewing.



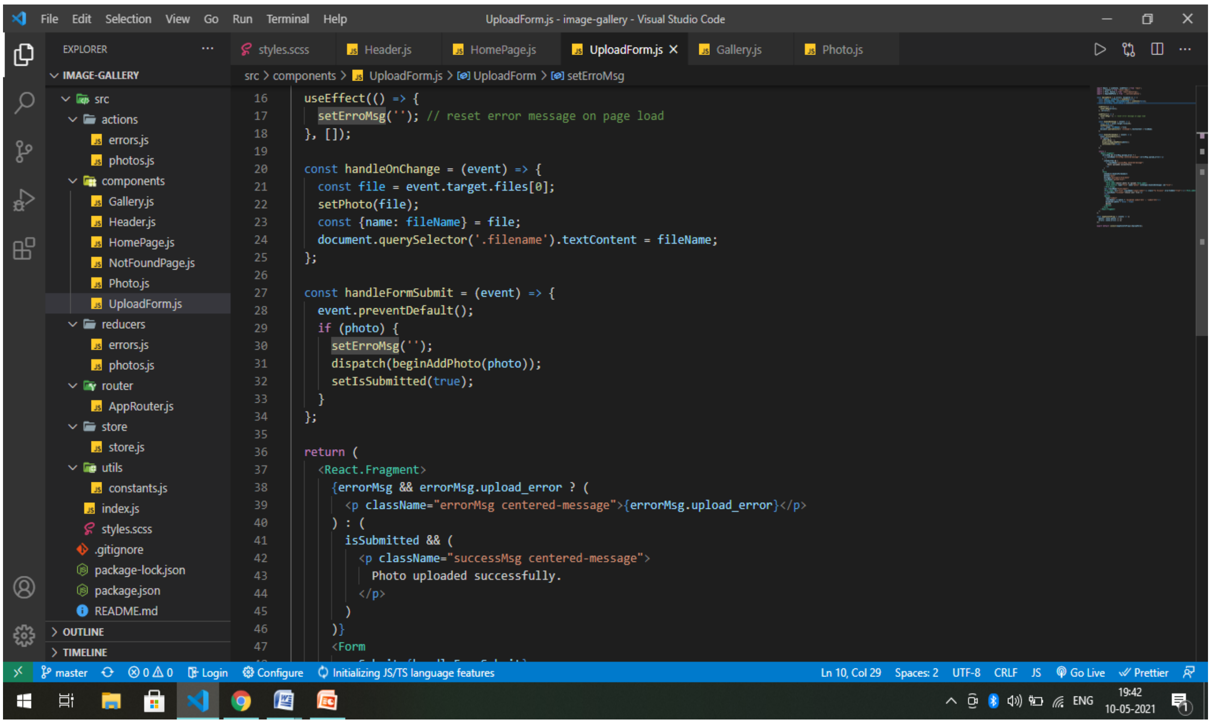
Code Screenshots:

1. We have created several components in our application that are connected well like this is the Header.js component where we have rendered our main page. In this page we have added a header with project name and also a navigation bar to work through the modules of the project.



```
import React from 'react';
import { NavLink } from 'react-router-dom';
import { FontAwesomeIcon } from '@fortawesome/react-fontawesome';


const Header = () => (
  <header>
    <div className="navigation">
      <div className="logo">
        <h1>Picsgram</h1>
      </div>
      <button className="icon" >
      <i class="fa fa-bars"></i>
      </button>
    </div>
    <h1>Photo Gallery App</h1>
    <div className="links">
      <NavLink to="/" className="link">
        Home
      </NavLink>
      <NavLink to="/gallery" className="link">
        Gallery
      </NavLink>
    </div>
  </header>
);

export default Header;
```

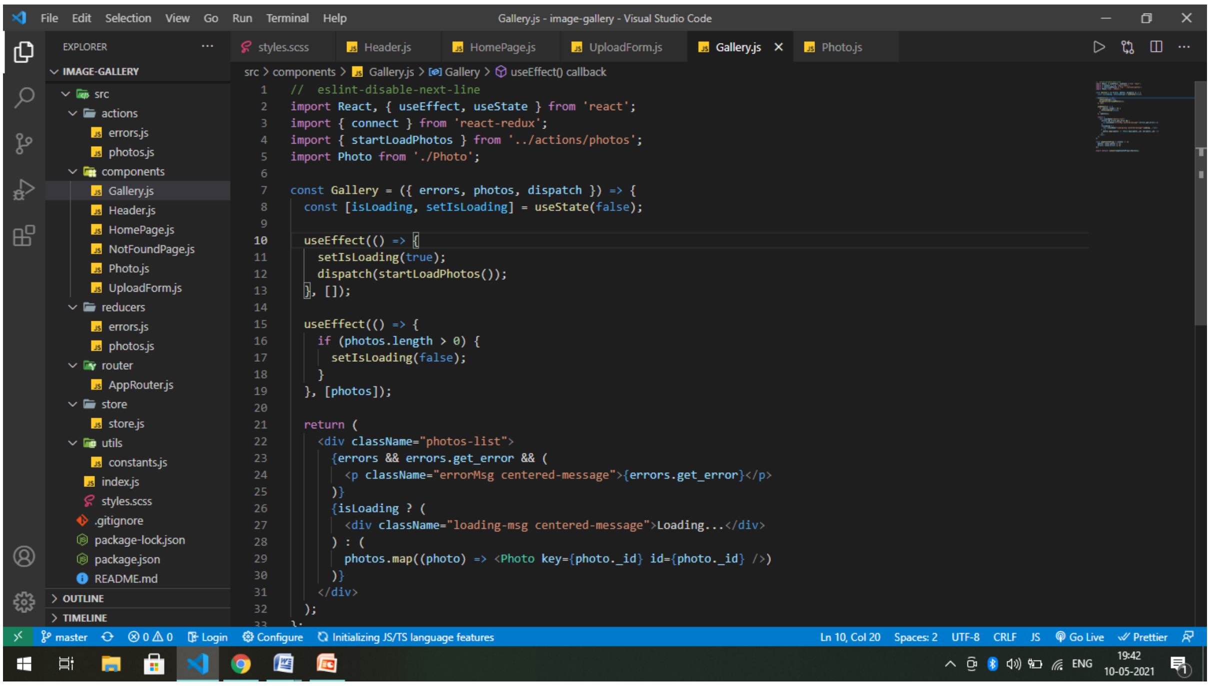
2. We have created UploadForm.js component where we have created a form to take the input image from the user with all the validations and connecting the user interface with the backend also.



UploadForm.js - image-gallery - Visual Studio Code
src > components > UploadForm.js > UploadForm > setErroMsg
Ln 10, Col 29

```
  useEffect(() => {
    setErroMsg(''); // reset error message on page load
  }, []);

  const handleOnChange = (event) => {
    const file = event.target.files[0];
    setPhoto(file);
    const {name: fileName} = file;
    document.querySelector('.filename').textContent = fileName;
  };

  const handleFormSubmit = (event) => {
    event.preventDefault();
    if (photo) {
      setErroMsg('');
      dispatch(beginAddPhoto(photo));
      setIsSubmitted(true);
    }
  };

  return (
    <React.Fragment>
      {errorMsg && errorMsg.upload_error ? (
        <p className="errorMsg centered-message">{errorMsg.upload_error}</p>
      ) : (
        isSubmitted && (
          <p className="successMsg centered-message">
            Photo uploaded successfully.
          </p>
        )
      )}
      <Form
```

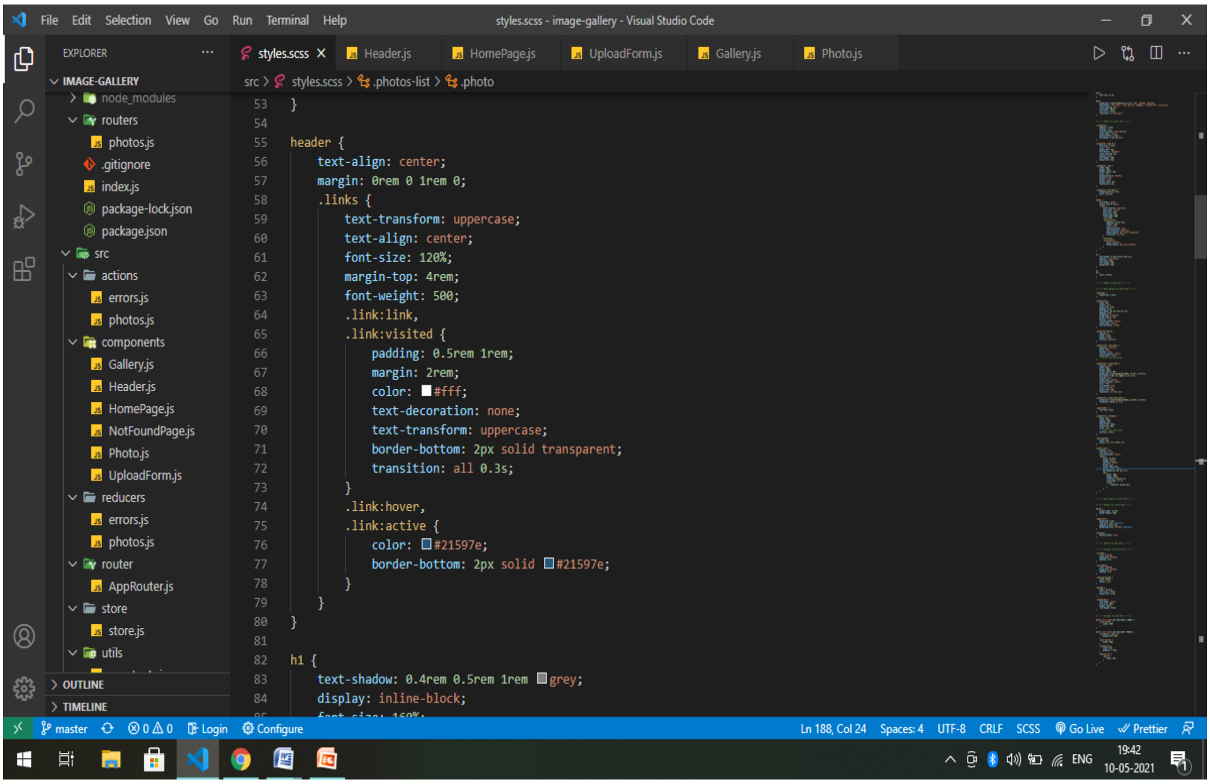
3. We have created a Gallery.js component where we render other component inside it and design it where the images got saved after extracting from the database.



Gallery.js - image-gallery - Visual Studio Code
src > components > Gallery.js > Gallery > useEffect() callback
Ln 10, Col 20

```
//  eslint-disable-next-line
import React, { useEffect, useState } from 'react';
import { connect } from 'react-redux';
import { startLoadPhotos } from '../actions/photos';
import Photo from './Photo';

const Gallery = ({ errors, photos, dispatch }) => {
  const [isLoading, setIsLoading] = useState(false);

  useEffect(() => {
    setIsLoading(true);
    dispatch(startLoadPhotos());
  }, []);

  useEffect(() => {
    if (photos.length > 0) {
      setIsLoading(false);
    }
  }, [photos]);

  return (
    <div className="photos-list">
      {errors && errors.get_error && (
        <p className="errorMsg centered-message">{errors.get_error}</p>
      )}
      {isLoading ? (
        <div className="loading-msg centered-message">Loading...</div>
      ) : (
        photos.map((photo) => <Photo key={photo._id} id={photo._id} />)
      )}
    </div>
  );
```

4. We have created a Photo.js component where we created a card format to display the images when connected to server.



```scss
}

header {
    text-align: center;
    margin: 0rem 0 1rem 0;
    .links {
        text-transform: uppercase;
        text-align: center;
        font-size: 120%;
        margin-top: 4rem;
        font-weight: 500;
        .link:link,
        .link:visited {
            padding: 0.5rem 1rem;
            margin: 2rem;
            color: #fff;
            text-decoration: none;
            text-transform: uppercase;
            border-bottom: 2px solid transparent;
            transition: all 0.3s;
        }
        .link:hover,
        .link:active {
            color: #21597e;
            border-bottom: 2px solid #21597e;
        }
    }
}

h1 {
    text-shadow: 0.4rem 0.5rem 1rem grey;
    display: inline-block;
    font-size: 160%;
```

Conclusion

Image uploaders have a variety of use cases ranging from file uploads on forms to updating your social media profile. React based application is far more interactive and flexible in every perspective. This application will provide great features for the users who are illustrators or photographers or designers. It will help the in getting the images in handy and can use them in their projects and take ideas from them to enhance their skills too. Hence we conclude that this application will provide a clear and good visual presentation to the user.

Online Git repository

Link of project github repository: https://github.com/nidhijaincs18/image-gallery

Reference

MongoDB - Wikipedia
https://en.wikipedia.org › wiki › MongoDB

Express.js - Wikipedia
https://en.wikipedia.org › wiki › Express

React (JavaScript library) - Wikipedia
https://en.wikipedia.org › wiki › React_(JavaScript_libr...

ReactJS - Overview - Tutorialspoint
https://www.tutorialspoint.com › reactjs › reactjs_overvi...

Visual Studio Code - Wikipedia
https://en.wikipedia.org › wiki › Visual_Studio_Code