

GLA University, Mathura

Full Stack-II Project

Project Review Synopsis



Project Title: “Picsgram- An image uploader”

Team Members

Names:

- | | |
|-------------------------|-------------|
| 1. Gauri Agrawal(97) | [Section-A] |
| 2. Nidhi Jain (37) | [Section-D] |
| 3. Radhika Bansal (44) | [Section-D] |
| 4. Sakshi Bhardwaj (53) | [Section-D] |

Submitted To: Mr. Pankaj Kapoor

Index

1. Project Introduction	1
2. Problem Statement	1
3. Working Methodology	2
4. Module Description	3
5. Softwares Used	4
6. Scope of the Project	4
7. Online Git Repository	4
8. Conclusion	5

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 promise-based 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 battle-tested and matured frontend frameworks in the world and

express.js is its 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 server-side 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

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.

Online Git repository

Link of project github repository:

<https://github.com/nidhijaincs18/image-gallery>

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.