

Web Based Photo Analysis and Learning

Sohel Hossain¹, KrishnenduSaha^{1*}, Prayasee Ghosh¹, Partha Pratim Dasgupta²

¹Student, department of computational science, Brainware University, West Bengal, India

²Department of computational science, Brainware University, West Bengal, India

*saha.krishnendu1998@gmail.com

*Corresponding Author

Abstract

Photography is just like art; it is an art of capturing a moment to make it memorable and special, not only perfect. We have to appreciate these moment catchers, photographers, and their talents. In this scenario, we know that social media is an active medium for connecting with these talents. But we all may have little or vast knowledge about the redundancy of social media. It just becomes empty rather than a learning medium, where we spend our valuable time just for nothing or for filtering something from those nothings and just unsighted these talents. So, we are here with an application where we are going to figure out every user's choice and filter out these unnecessary topics so that the user can praise these talents with an open mind and help them to move forward. Here photographer can reach to other photographers, shares their knowledge, and gather some experience to enhance their talents. And the dream to make this social media a learning medium will become true.

Keywords: Photo Analysis and Learning, MERN Stack, MongoDB-Express-React-Node, MVC Format, Sentimental Analysis

1. Introduction

PKS is a photography website, starts development in November 2020, and it's still on-going. Here photographers can show their daily activities. In this website we have covered some basic features like home page, rather we can say a gallery, log in for specific & personalized access, user-specific own gallery, an “about us” page where users can find their own or the other users’ information. We have also provided an “admin portal” which can be accessed by admin when they login to the website with their specific credentials. Guests users who haven’t created an account won’t be able to view their profile, they can only use the basic features of the website. For building this website we use the MERN stack, but at the basic level, we just use MySQL, rather using MongoDB, as our database schema. In this project, ReactJS is used for the frontend, Express (NodeJS) for the backend part.

2. Work-flow

MERN stack, MongoDB-Express-React-Node these are some of the well-known stacks all over the developing sector. So, here in our project, we are trying to implement these stacks to make this site more efficient and quickerto give a better experience for our users. Here we also use the MySQL server to store the user credential data. When the users open the web app they can see a fully dynamic ReactJS user-interface, where they may log in or not, but they can view the posts and portfolios of other users. They also can do some posts, give their votes and suggestions to remains only if they sign-up with our application. After doing these, the credential data will be reached to our backend Express server and stored in our MySQL database.

2.1. ReactJS:

ReactJS is the front stack, in other word main user interface of our project, will help to show the pictures and handle the visual related part. In this we use custom JSX elements, Java-Script for client-side logical part. Through ReactJS this web application is created like one-page simulation

app. Only when the user tries to log in or create his account the website will refresh otherwise it's work like and application. With ReactJS we hit a specific API with specific JSON data for fetch data from database. And also, we don't need to refresh the page to load more pictures or data though it's automatically loads it for us and when someone click on load more the data shown instantly. It reduces the loading time of the page and though all data loading while surfing so we can display full quality images straight forward.

2.2. NodeJS:

NodeJS is the backend stack of our project, will help the frontend, ReactJS part, with giving some of the users' credential data to authenticate the user at the login section. NodeJS mainly uses the MVC format but, we are creating a rest API with JWT token authorization instead of any view section. When the frontend (ReactJS part) request for data via API in backend (Express) it will authenticate the token verification (if it is restricted, else it will return data) return desired data in JSON format, by which the frontend views will be modified as per the requirement.

2.3. Database:

Here we are using SQL and NoSQL both type of database schema as we are using MySQL for user management and other relation type data driven work. And for storing the unstructured document type data we are using MongoDB. For MongoDB we are using Mongoose driver in our backend model and for MySQL we are using MySQLi driver with eloquent/pdo queries.

2.4. Responsiveness:

This website is a basically a web app so we mainly focus on Features, Simple User Interface and Devise based Responsiveness. Responsiveness is the main pillar of this web app and we designed it such a way that everyone can use the full potentiality of this website. We arrange the pictures such a way that everyone can see what a photographer want us to see.

2.5. User interface:

Though it is a Web Application so we focused on Simple web-based interface with all features that can be possible to implement within it. It is a mainly one page web app that only refresh some time when a user try to log in or vice versa.

2.6. Features:

- **Unique User Account:** A user can create his fully personalized account like various types of social media platforms. In this, a user can add, hide and edit their personal details after creating his account. Creating an account will give that user the access of posting their daily photographic activities and will able to vote and comment on posts.
- **Guest User Account:** A user can see a very few selected type of photos without creating an account in our web app. But, will not be able to post, comment and vote on any of the selected or other posts.
- **High Resolution Pictures:** A user can upload a High resolution image and edit it online. The image and the edited image will store simultaneously. This web application show this images in it's perfect aspect ratio and a user can see the zoom view by clicking on the image in gallery.

3. Future scope

- In future we will add a voting system in which we will calculate the user profile activity points and based on these points we give them ranking accordingly with a proper algorithm.
- According to the profile ranking user's photo will be shown in the trending section.
- The post will be filtered according to the user's choice and interest with a particular algorithm.
- Filtering out the futile comments by sentimental analysis, we are going to deliver real admiration to our new users so that they can recognize their talent and raise that.
- To sustain the users' post courtesy our website maintains some policy (1) if any of our user (guest or permanent) download any photos from our site there must be a copyright mark of the original user (2) profile view and the download notification will arrive to the primary user.

References

1. Burgin, V. ed., 1982. *Thinking photography*. Macmillan International Higher Education.
2. Peterson, B., 2015. *Learning to see creatively: Design, color, and composition in photography*. Amphoto Books.
3. Yin, W., Mei, T., Chen, C.W. and Li, S., 2013. Socialized mobile photography: Learning to photograph with social context via mobile devices. *IEEE Transactions on Multimedia*, 16(1), pp.184-200.
4. Malu, G., Bapi, R. S., & Indurkha, B. (2017). Learning photography aesthetics with deep cnns. *arXiv preprint arXiv:1707.03981*.
5. Ning, H., Li, Z., Hodgson, M.E. and Wang, C.S., 2020. Prototyping a social media flooding photo screening system based on deep learning. *ISPRS International Journal of Geo-Information*, 9(2), p.104.
6. Yan, Z., Zhang, H., Wang, B., Paris, S. and Yu, Y., 2016. Automatic photo adjustment using deep neural networks. *ACM Transactions on Graphics (TOG)*, 35(2), pp.1-15.
7. Stone, D., Jarrett, C., Woodroffe, M. and Minocha, S., 2005. *User interface design and evaluation*. Elsevier.
8. Aryal, S., 2020. MERN stack with modern web practices.
9. Hounsfield, G.N., 1976. Picture quality of computed tomography. *American journal of Roentgenology*, 127(1), pp.3-9.
10. Hahn, E., 2016. *Express in Action: Writing, building, and testing Node.js applications*. Manning Publications.