VITTERSTOCK
Photo Sharing App

A PROJECT REPORT

Submitted in partial fulfillment for the award of the degree of
B.Tech

in

Information Technology

By

16BIT0044	Pranshu Pranjal
16BIT0133	Ekansh Gayakwad
16BIT0162	Manas Dange

Under the guidance of
Prof. Sumaiya Thaseen

VIT®
UNIVERSITY
(Estd. u/s 3 of UGC Act 1956)
VELLORE ▪ CHENNAI
www.vit.ac.in

NOVEMBER, 2017

School of Information Technology and Engineering

CERTIFICATE

This is to certify that this project report titled "**VITTERSTOCK Photo Sharing App**" is the bonafide work of **PRANSHU PRANJAL (16BIT0044), EKANSH GAYAKWAD (16BIT0133)** and **MANAS DANGE (16BIT0162)** who carried out the project work under my supervision.

(SIGNATURE OF THE SUPERVISOR)

ACKNOWLEDGEMENT

This project was very interesting, informative but challenging to make. We are deeply indebted to all those who helped us in completion of this work. We are thankful to our Professor Sumaiya Thaseen, for her support, guidance and help throughout the project and without her we would have not been able to complete this project.

We would like to thank VIT University to provide us facilities and teachers to guide us throughout our work.

Manas Dange

(16BIT0162)

Pranshu Pranjal
(16BIT0144)

Ekansh Gayakwad
(16BIT0133)

TABLE OF CONTENTS

EXECUTIVE SUMMARY

The past decade has witnessed a sudden boom in social network based applications. VITTERSTOCK is a medium which allows a user to share images with other users, and users can download images shared by others users. It has been developed keeping in mind the current trend of taking quality photos and sharing them due to boost in recent mobile technology. It has been developed utilising the latest web technology like Node.js, MongoDB, Express Framework, Bootstrap, Font Awesome, CSS Animations, HTML 5 etc.

LIST OF FIGURES

INTRODUCTION

The application that we have proposed to develop here is based on the model of photo sharing social network for amateur or even professional photography enthusiasts. Here, users will be able to interact with the photos shared by their peers in three ways- view, like and comment.

VITTERSTOCK, the application we committed to develop is completely web-based. Being a web-app, it makes the use of HTML5 for page structure, CSS for page styling, we also used javascript to increase the user-interactivity of the application –these being the major front-end technologies. For back-end development, VITTERSTOCK uses node.js (a javascript based framework) for server side scripting and mongoDB to store the data collected from the users through the application. The prime objective of "VITTERSTOCK" is to create a full-fledged website which enables users to upload and download pictures, and share the picture publicly. The user not only can share photo but it also enables users to follow other users, like other user's post and comment on other's images.

This will allow people to flaunt their photographic skills and will provide a potential platform for budding photographers. Moreover, the project would be a famous platform for different organisations to host competitions and scourge hidden talents.

Moreover, third party web frameworks like Bootstrap (for styling), jQuery (a javasript library, for easing up the process of writing front-end scripts), passport.js (to authenticate user credentials), BCrypt (to encrypt sensitive data) are used in our photo sharing app.

SCOPE AND OBJECTIVES

The prime objective of “VITTERSTOCK” is to create a full-fledged web application which enables users to view pictures, and share the picture publicly. The user not only can share photo but it also enables users to follow other users, like other user’s post and comment on other’s images.

This will allow people to flaunt their photographic skills and will provide a potential platform for budding photographers.

It will also be useful for flaunting the beauty of VIT Campus and will attract attention of freshers.

Moreover, the project would be a famous platform for photography club which could host competitions and scourge hidden talents.

LITERATURE SURVEY

TITLE: “A Survey Paper on Photo Sharing and Privacy Control Decisions”

Author: Divyalaxmi R. Nampalli, Prof. Trupti Dange

Abstract:

Photo sharing is an alluring component which enhances Online Social Networks. Sadly, it may release clients' security on the off chance that they are permitted to post, remark, and label a photograph openly. The situation when a client shares a photograph containing people other than her (termed co-photograph for short) have been thoroughly discussed. The need to minimize the security breaches that happen because posting the photos of people without the awareness of people involved in photo. For this reason, we require a proficient facial acknowledgment (FR) framework that can perceive everybody in the photograph. Notwithstanding, all the more requesting security setting may restrain the photographs' quantity freely accessible to prepare the FR framework. To manage this issue, latest growth in machine learning endeavors to use clients' private photographs to plan a customized FR framework particularly prepared to separate conceivable photograph co-proprietors without releasing their protection. Also disseminated accords based system to diminish the computational many-sided quality and ensure the private preparing set are widely in use. The machine learning framework is better than other conceivable methodologies as far as acknowledgment proportion and effectiveness is concerned. Their instrument is executed as a proof of idea Android application on Facebook's stage. OSNs will not contaminate to true users and polluted by unauthorized users and their posting the photos in unsecure way. Hence OSNs will be secure and safest.

TITLE: "COMMUNITY PHOTO SHARING: MOTIVATIONAL AND STRUCTURAL ANTECEDENTS"

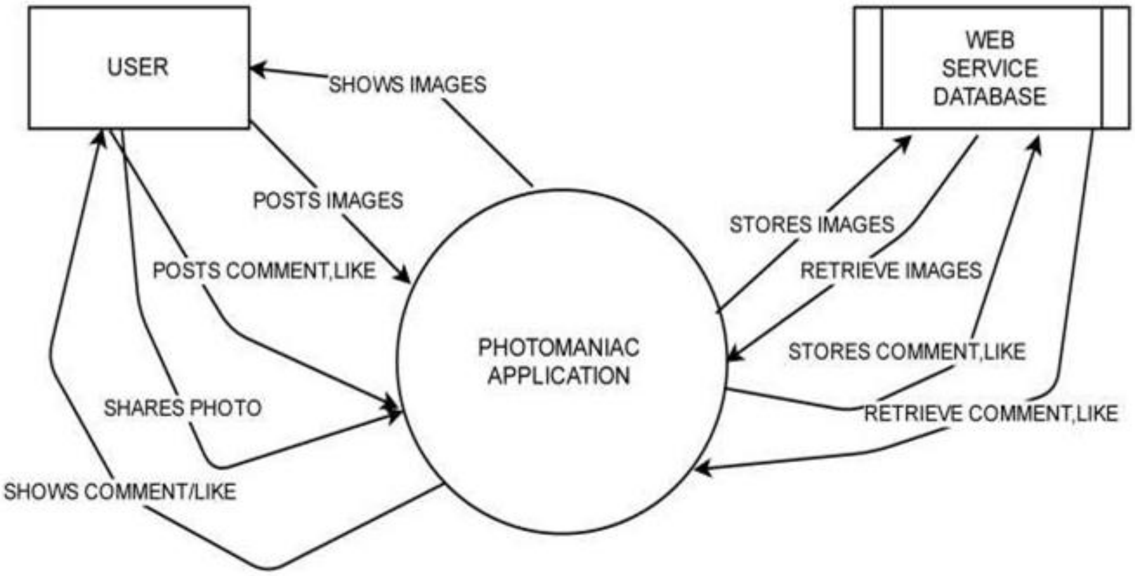
Author: Oded Nov, Chen Ye

Abstract:

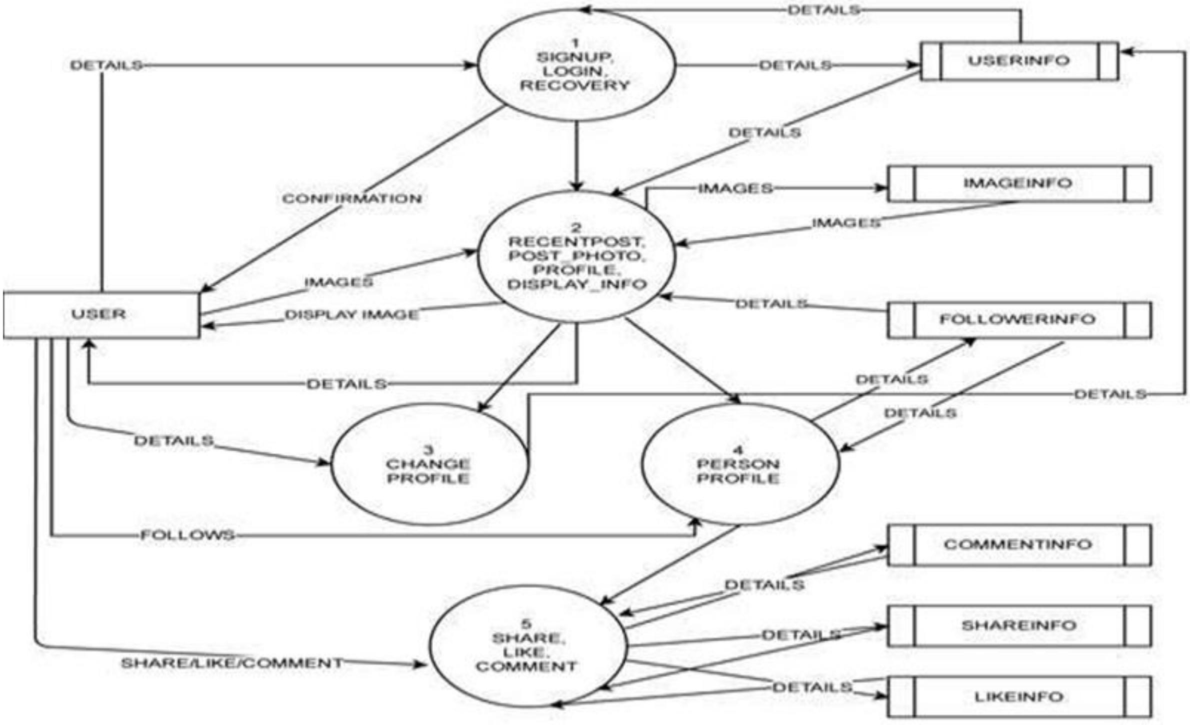
Findings from this study have several implications for theory and practice: first, the study contributes to the body of information systems literature on user contribution by developing a framework to help understand users' motivations to contribute in a new IS-facilitated phenomenon – social computing and in particular online photo sharing. User motivation to contribute has been studied in other social collectives such as online knowledge sharing communities. Social computing services such as Flickr and YouTube enable a vast number of individual users to be contributors and users of content. Therefore, researchers also need to take a user-centric approach in understanding the dynamics of content contribution in these social computing environments. In this study, the role of individual motivations and structural embeddedness in user contribution was illustrated. This understanding may serve as a useful first step for further research of social computing phenomena. The contribution of this work also points for directions for further research: the decoupling of the "first act" of creation from the "second act" of sharing appears to have consequences that separate photo sharing from other forms of sharing, and requires further exploration. The second contribution of this study is in exploring the relationship between the self-development motivation and the amount photos shared: the negative association found warrants further research on the tradeoff between quantity and quality of contribution in a social computing context. A relevant yet distinct question is what motivates users to increase the quality of their contribution, and future development of this research project involves investigating the antecedents of contribution quality.

SOFTWARE DESIGN

HIGH LEVEL DESIGN DIAGRAM



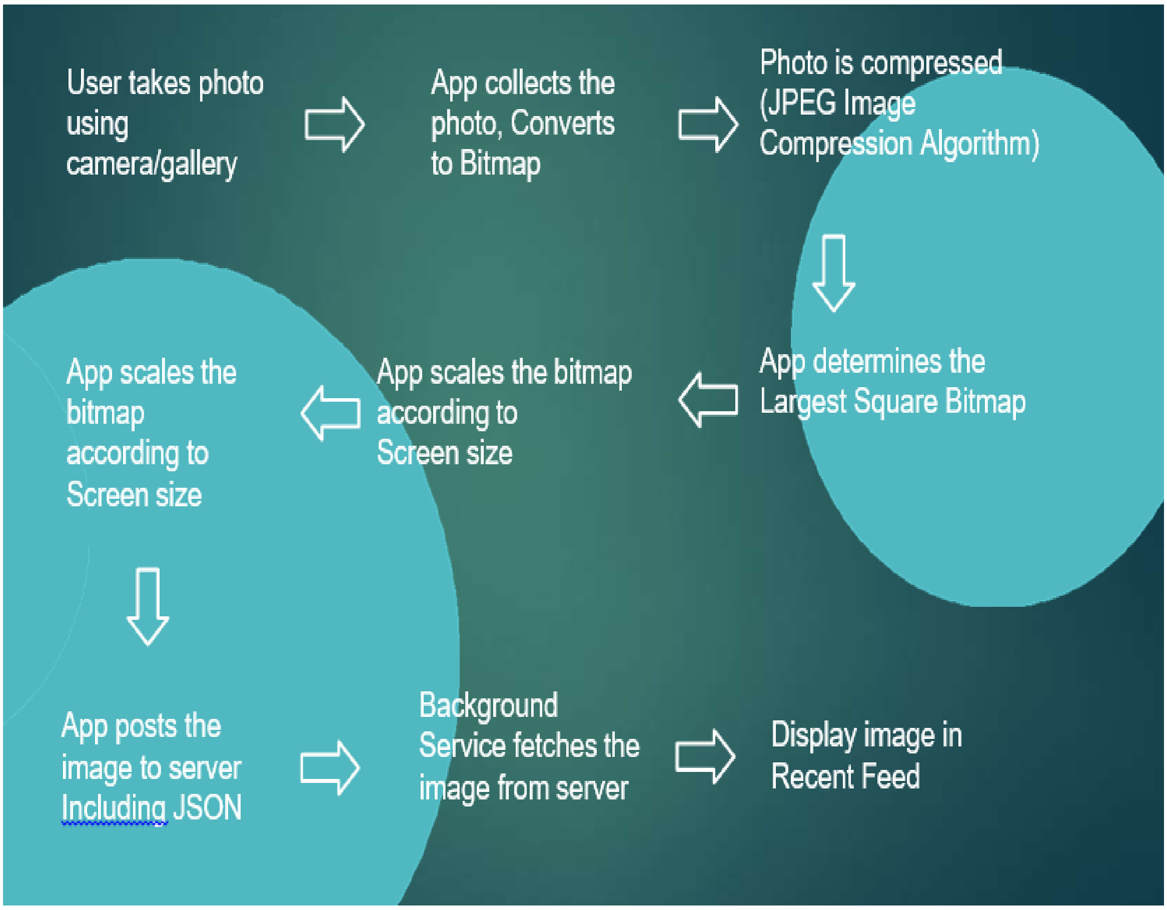
LOW LEVEL DESIGN



FEATURES:

- Easy to use and Intuitive User Interface
- Various Password recovery scheme
- Allows user to upload from existing photos
- Allows user to follow other user
- Enables user to comment, like on other photos

METHODOLOGY



User takes photo using camera/gallery

App collects the photo, Converts to Bitmap

Photo is compressed (JPEG Image Compression Algorithm)

App determines the Largest Square Bitmap

App scales the bitmap according to Screen size

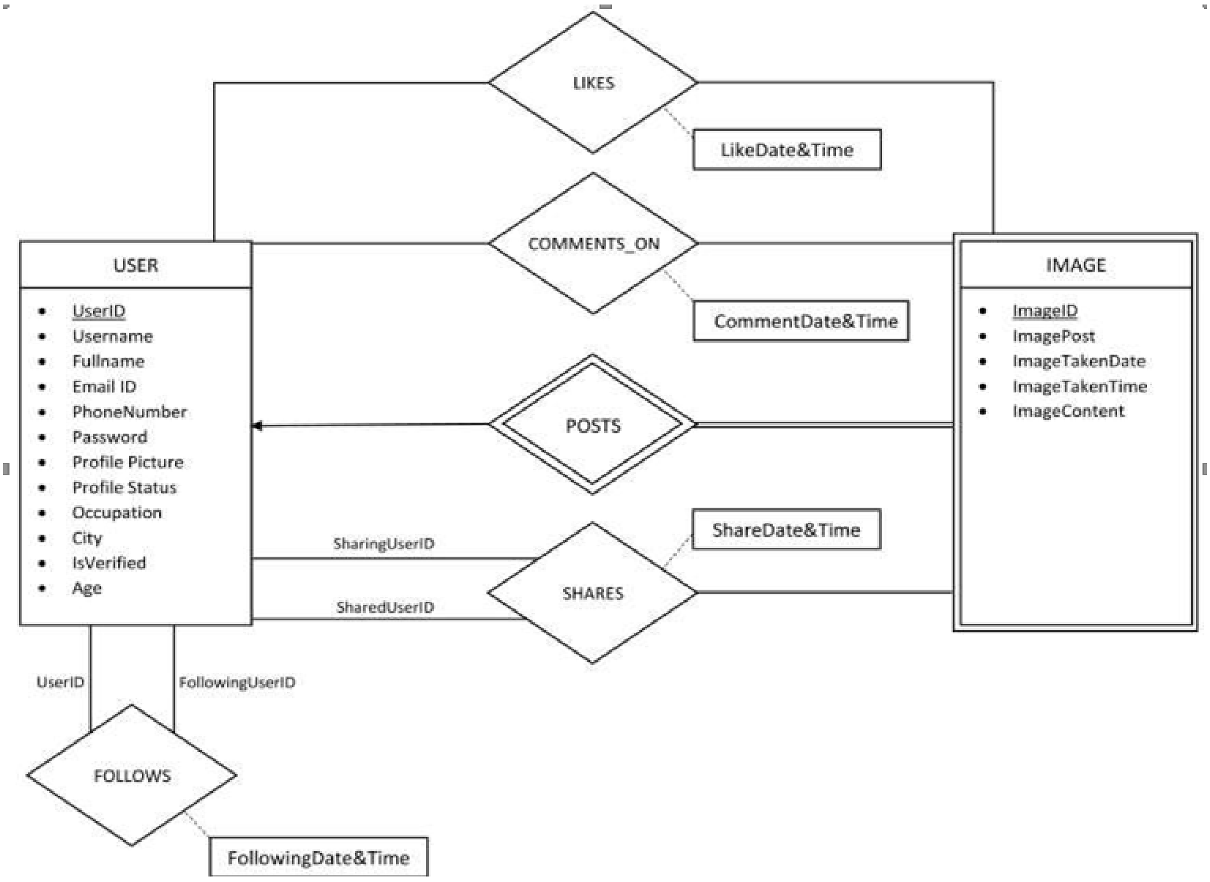
App scales the bitmap according to Screen size

App posts the image to server Including JSON

Background Service fetches the image from server

Display image in Recent Feed

ENTITY RELATIONSHIP DIAGRAM



SCHEMAS USED-

For Photos-

```
const
flag =
new
schema(
{
               photo:{
                   type:'string'
                   required:true
               },
               title:{
                   type:'string',
                   required:true
               },
    keywords:{
        type:'array'
        required: true
    },
    comments:{
        type:'array'
        default: [];
    },
    price:{
        type:'string'
        default: '0';
    },
    likes:{
        type:'number',
        default: 0;
    }
});
```

For User:

```
const flag = new schema({
     username:{
          type:'string',
          required:true,
          unique:true
     },
     password:{
          type:'string'
     },
     count:{
          type:'number',
          default:0
     },
     card:{
          cardnum:{
               type:'string',
          },
        amount:{
            type:'number'
        }
    },
    buy:{
        type:'array',
        default:[]
    }
});
```

Important Routes

To Upload Photo:

```
router.post('/add',(req,res)=>{
   var temp_path = req.files.upload.path;
   var date = new Date();
   var name = req.files.upload.name;
   var index = req.files.upload.name.indexOf('.');
   name = name.slice(index,name.length);
   var flag = date.getTime().toString()
   var path = p.join(__dirname + '/../public/uploads/' + flag +
name);
   console.log(path);
   var fs = require('fs');
   fs.rename(temp_path,path,(err)=>{
        if(err) return res.send(err);
        Photo.create({title:req.body.title}).then((result)=>{
            result.price = req.body.price;
            result.photo = '/uploads/' + flag + name;
            result.save().then((result)=>{
              console.log(req.session.user.username);
User.findOne({username:req.session.user.username}).then((result
)=>{
                console.log(result);
                result.count += 1;
                result.save();
                res.send('Uploaded.');
              });
            });
        });
   });
});
```

To Buy Photo:

```
router.get('/buy/:path',(req,res)=>{
  var amt;
  var path;
  Photo.findById(req.params.path).then((result)=>{
    console.log(result);
    amt = result.price;
    path = result.photo;
    return User.findById(req.session.user._id);
  }).then((result)=>{
    console.log(amt);
    if(result.card.amount < parseInt(amt)){
      return res.send("Cannot Buy, Insufficient Funds: $" +
result.card.amount + '.');
    }
    console.log(result);
    result.card.amount -= parseInt(amt);
    console.log(result.card.amount);
    result.buy.push(path);
    result.save();
    console.log(result);
    return res.send(“Bought.”);
  }).catch((err)=>{
    console.log(err);
    res.send(err);
  });
});
```

Create User

```
  User.create({username:req.body.username},(err,user)=>{
        if(err){
            console.log(err);
        }
        user.password = req.body.password;
        user.card.cardnum = req.body.cardnum;
        user.card.amount = parseInt(req.body.amount);
        user.markModified('card');
        console.log(user.card);
        user.save();
        user.hashandsave().then((msg)=>{
            req.session.user = user;
            console.log(user);
            console.log('User Created.')
        }).catch((err)=>{
            console.log('Cannot Create.');
        });
        console.log(user);
    });
```

```
});
```

Login Router

```
router.post('/login',(req,res)=>{
   User.compare(req.body.username,req.body.password,
(err,res,user)=>{
    if(err){
        console.log("Cannot Login.");
    }
    else{
        req.session.authenticated = true;
        req.session.user = user;
        console.log("LoggedIn.");
        res.send("logged in");
        res.render('index');
        // res.redirect("www.google.com");
    }
   });
});
```

Logout Router

```
router.get('/logout',(req,res)=>{

    if(req.session.authenticated){
        delete req.session.authenticated;
        delete req.session.user;
    }
    console.log("Logged Out.");
});
```

MODULES USED

BCRYPT

Bcrypt is a password hashing function designed by Niels Provos and David Mazières, based on the Blowfish cipher, and presented at USENIX in 1999. Besides incorporating a salt to protect against rainbow table attacks, bcrypt is an adaptive function: over time, the iteration count can be increased to make it slower, so it remains resistant to brute-force search attacks even with increasing computation power.

The bcrypt function is the default password hash algorithm for OpenBSD and other systems including some Linux distributions such as SUSE Linux. The prefix "$2a$" or "$2b$" (or "$2y$") in a hash string in a shadow password file indicates that hash string is a bcrypt hash in modular crypt format. The rest of the hash string includes the cost parameter, a 128-bit salt (base-64 encoded as 22 characters), and 184 bits of the resulting hash value (base-64 encoded as 31 characters). The cost parameter specifies a key expansion iteration count as a power of two, which is an input to the crypt algorithm.

For example, the shadow password record

$2a$10$N9qo8uLOickgx2ZMRZoMyeIjZAgcfl7p92ldGxad68LJZdL17lhWy

specifies a cost parameter of 10, indicating 210 key expansion rounds.

The salt is N9qo8uLOickgx2ZMRZoMye and the resulting hash is IjZAgcfl7p92ldGxad68LJZdL17lhWy.

Per standard practice, the user's password itself is not stored.

NODE.JS

Node is similar in design to, and influenced by, systems like Ruby's Event Machine or Python's Twisted. Node takes the event model a bit further. It presents an event loop as a runtime construct instead of as a library. In other systems there is always a blocking call to start the event-loop. Typically behavior is defined through callbacks at the beginning of a script and at the end starts a server through a blocking call. In Node there is no such start-the-event-loop call. Node simply enters the event loop after executing the input script.

Node exits the event loop when there are no more callbacks to perform. This behavior is like browser JavaScript — the event loop is hidden from the user.

HTTP is a first class citizen in Node, designed with streaming and low latency in mind. This makes Node well suited for the foundation of a web library or framework.

Passport.js

Passport is authentication middleware for Node. It is designed to serve a singular purpose: authenticate requests. When writing modules, encapsulation is a virtue, so Passport delegates all other functionality to the application. This separation of concerns keeps code clean and maintainable, and makes Passport extremely easy to integrate into an application.

In modern web applications, authentication can take a variety of forms. Traditionally, users log in by providing a username and password. With the rise of social networking, single sign-on using an OAuth provider such as Facebook or Twitter has become a popular authentication method. Services that expose an API often require token-based credentials to protect access.

Passport recognizes that each application has unique authentication requirements. Authentication mechanisms, known as strategies, are packaged as individual modules. Applications can choose which strategies to employ, without creating unnecessary dependencies.

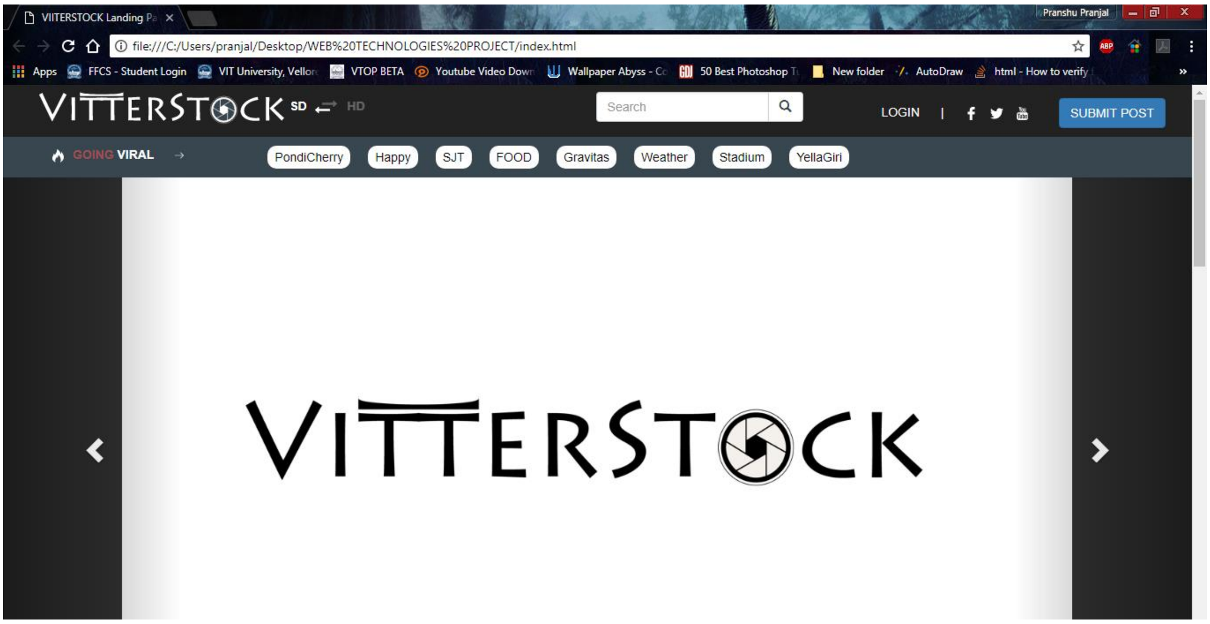
Express Photo Gallery

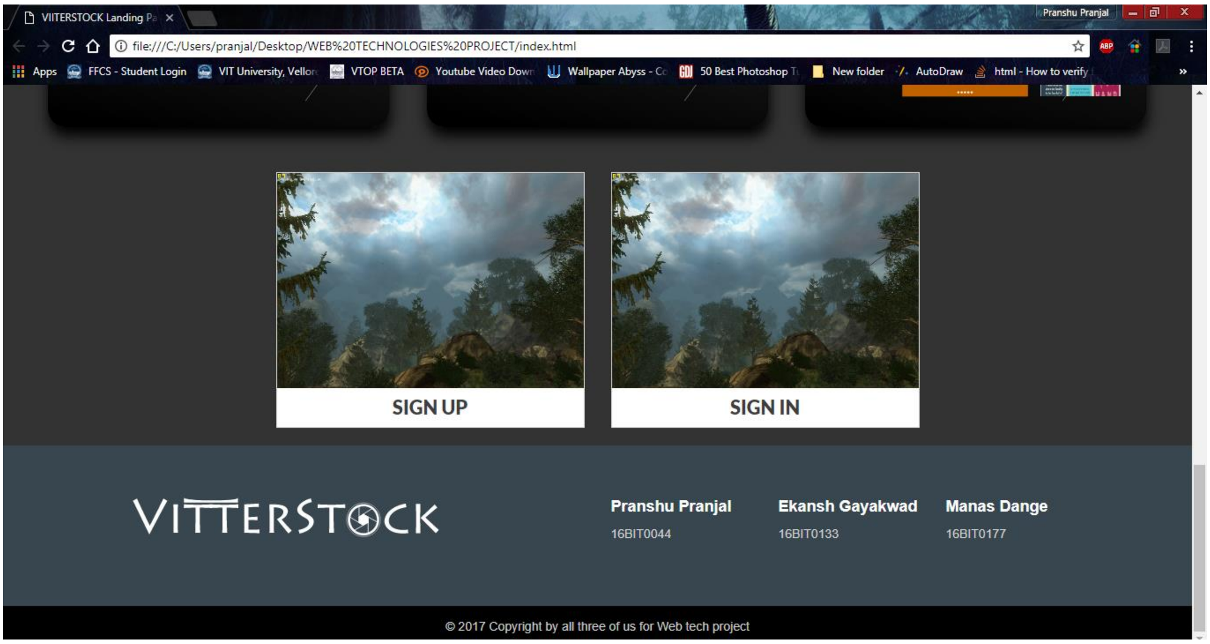
EPG (express-photo-gallery) is a node module that creates an Express.js middleware function for hosting stylish and responsive photo galleries using jQuery lightgallery.

EPG will automatically look through the provided directory for a thumbs subdirectory and previewssubdirectory, which should both contain files with the same filenames as those in the base folder. For example:

If the photo directory does not have thumbs or previews, EPG will host the images in the base directory, without displaying thumbnails.

SCREENSHOTS



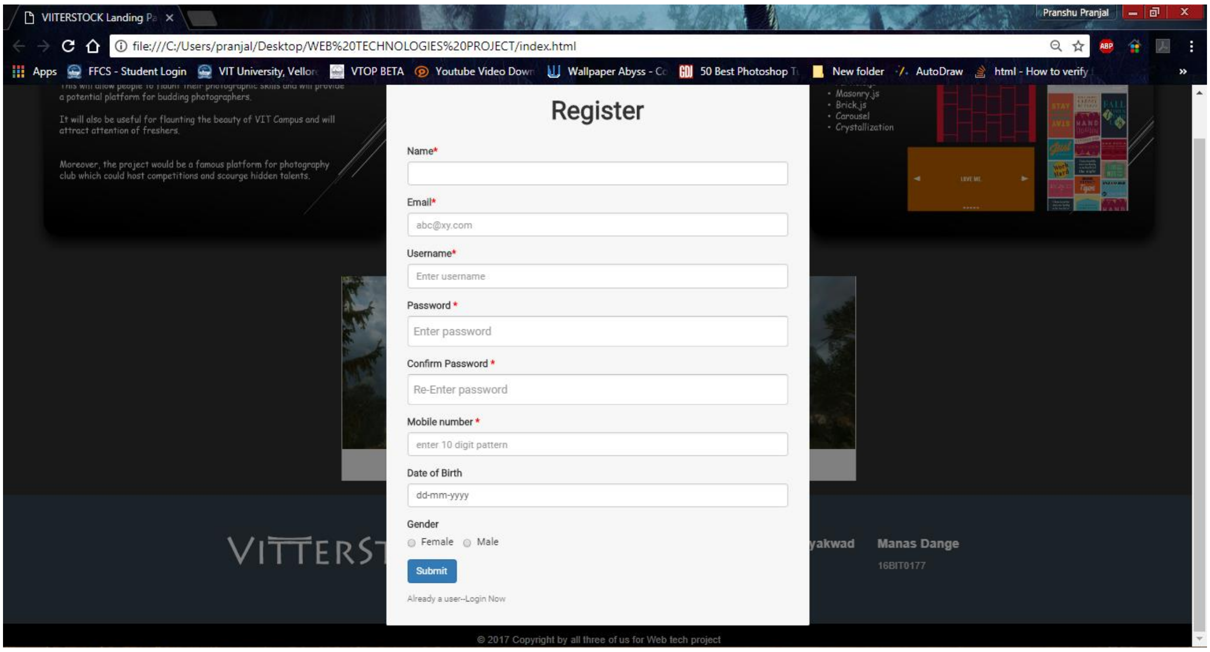

VITTERSTOCK Landing Pa
file:///C:/Users/pranjal/Desktop/WEB%20TECHNOLOGIES%20PROJECT/index.html
Pranshu Pranjal
SIGN UP
SIGN IN
VITTERSTOCK

Pranshu Pranjal
16BIT0044

Ekansh Gayakwad
16BIT0133

Manas Dange
16BIT0177

© 2017 Copyright by all three of us for Web tech project



This will allow people to flaunt their photographic skills and will provide a potential platform for budding photographers.

It will also be useful for flaunting the beauty of VIT Campus and will attract attention of freshers.

Moreover, the project would be a famous platform for photography club which could host competitions and scourge hidden talents.

- Masonry.js
- Brick.js
- Carousel
- Crystallization

Register

Name*

Email*
abc@xy.com

Username*
Enter username

Password *
Enter password

Confirm Password *
Re-Enter password

Mobile number *
enter 10 digit pattern

Date of Birth
dd-mm-yyyy

Gender
○ Female ○ Male

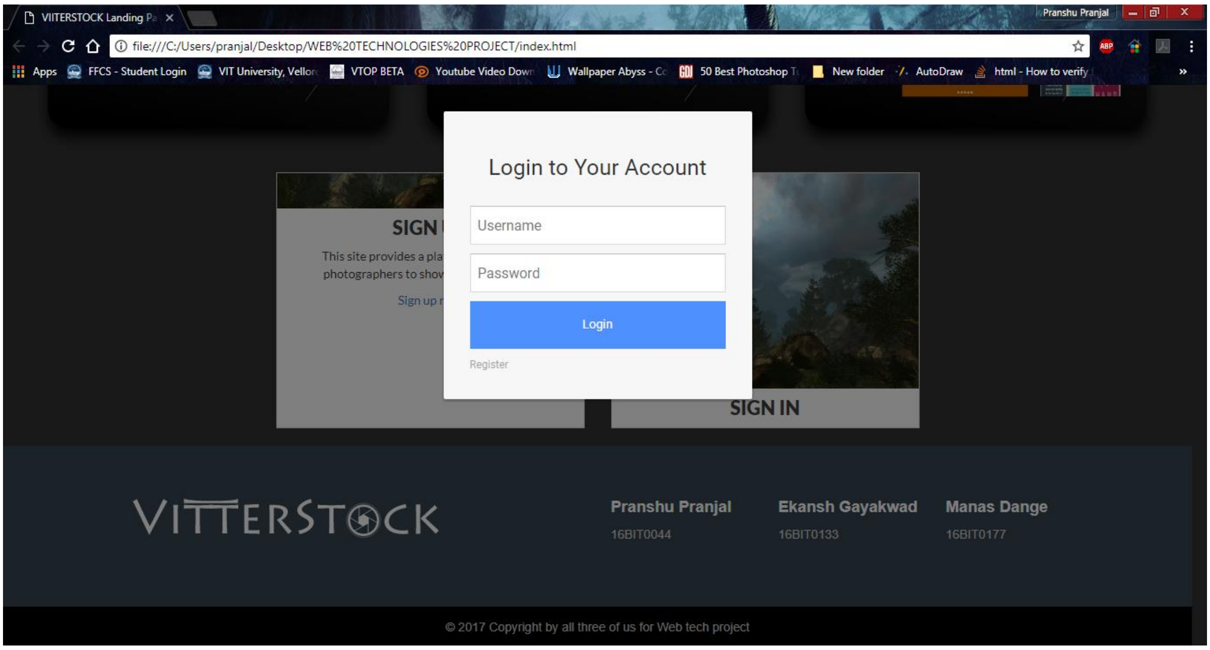
Submit

Already a user--Login Now

VITTERST

yakwad Manas Dange
16BIT0177

© 2017 Copyright by all three of us for Web tech project



SIGN

This site provides a pla
photographers to show

Sign up r

Login to Your Account

Username

Password

Login

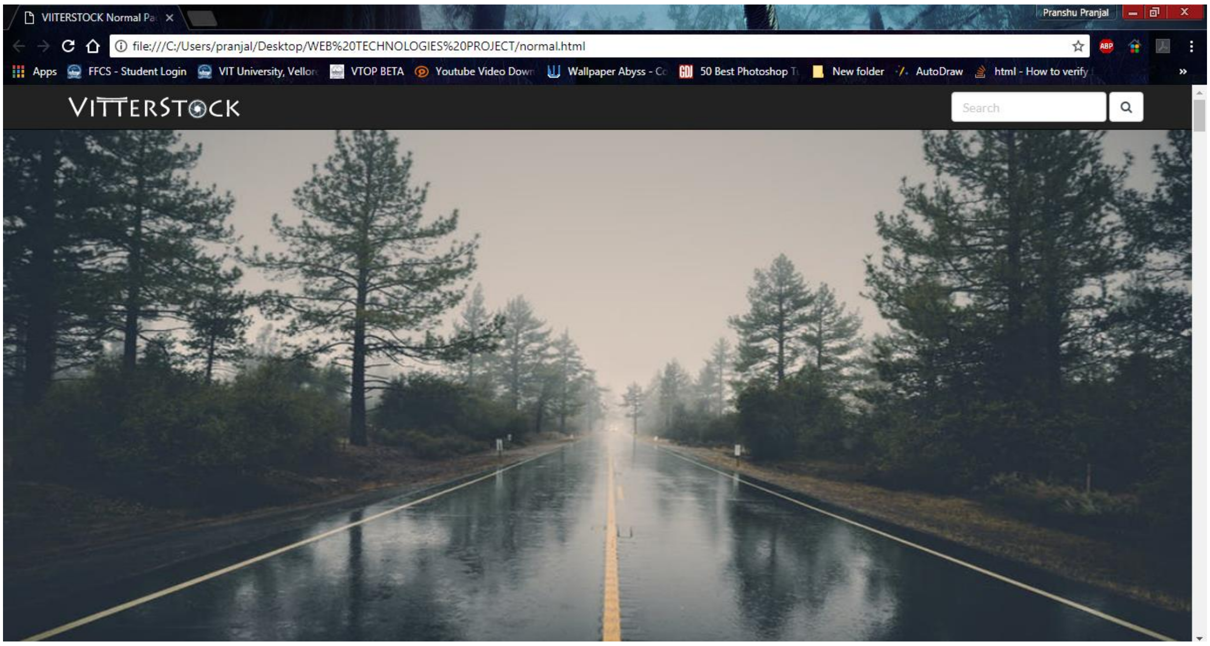
Register

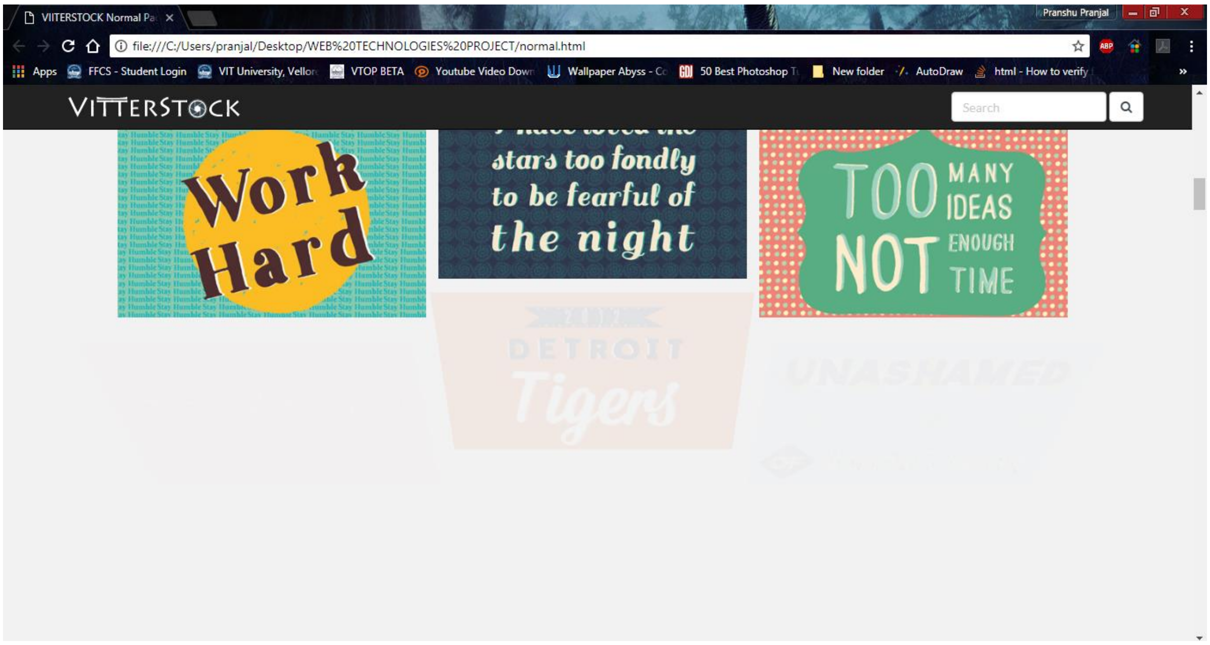
SIGN IN

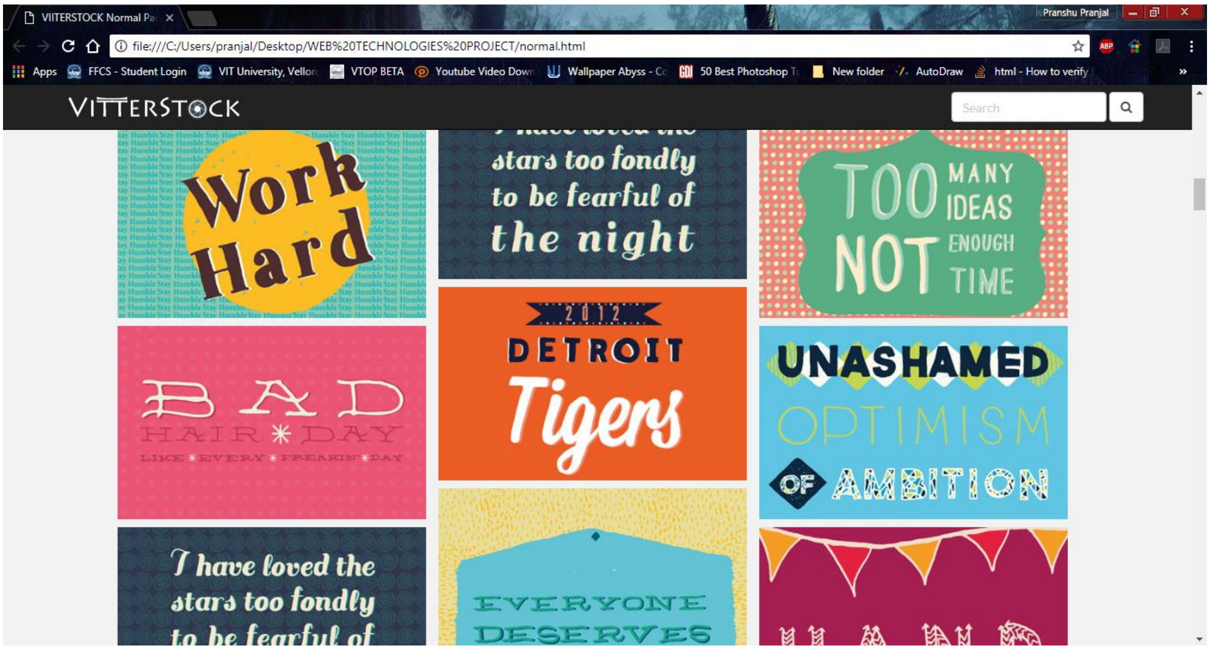
VITTERSTOCK

Pranshu Pranjal	Ekansh Gayakwad	Manas Dange
16BIT0044	16BIT0133	16BIT0177

© 2017 Copyright by all three of us for Web tech project


VIITERSTOCK Normal Pa
Pranshu Pranjal
file:///C:/Users/pranjal/Desktop/WEB%20TECHNOLOGIES%20PROJECT/normal.html
Apps
FFCS - Student Login
VIT University, Vellor
VTOP BETA
Youtube Video Down
Wallpaper Abyss - Co
50 Best Photoshop T
New folder
AutoDraw
html - How to verify
VITTERSTOCK
Search


VIITERSTOCK Normal Pa
Pranshu Pranjal
file:///C:/Users/pranjal/Desktop/WEB%20TECHNOLOGIES%20PROJECT/normal.html
Apps
FFCS - Student Login
VIT University, Vellor
VTOP BETA
Youtube Video Down
Wallpaper Abyss - Co
50 Best Photoshop T
New folder
AutoDraw
html - How to verify
VITTERSTOCK
Search
Work Hard
stars too fondly to be fearful of the night
TOO MANY IDEAS NOT ENOUGH TIME
DETROIT Tigers
UNASHAMED


VIITERSTOCK Normal Pa
Pranshu Pranjal
file:///C:/Users/pranjal/Desktop/WEB%20TECHNOLOGIES%20PROJECT/normal.html
Apps
FFCS - Student Login
VIT University, Vellor
VTOP BETA
Youtube Video Down
Wallpaper Abyss - Co
50 Best Photoshop Ti
New folder
AutoDraw
html - How to verify
VITTERSTOCK
Search
Work Hard
stars too fondly to be fearful of the night
TOO MANY IDEAS NOT ENOUGH TIME
BAD HAIR DAY LIKE EVERY FREAKIN DAY
2012 DETROIT Tigers
UNASHAMED OPTIMISM OF AMBITION
I have loved the stars too fondly to be fearful of
EVERYONE DESERVES

CONCLUSION

VITTERSTOCK provides a platform to search for images, share images, download them in high quality. It will be beneficial for free lancers looking forward to build their portfolio and provide a medium to showcase our photography skills. The VITTERSTOCK idea is an innovative one in providing a platform for the budding artists to get an exposure to different photography tricks by getting acquainted with the various type of photo uploaded by the users.

References

- https://stackoverflow.com/
- https://nodejs.org/en/
- https://www.pexels.com/
- http://fontawesome.io/
- https://www.w3schools.com/