ONLINE COMMUNITY MANAGEMENT SYSTEM

FINAL REVIEW REPORT

Submitted by

ASHWIN GOEL (17BCE2073) (L13+14)

AMITESH SRIVASTAVA (17BCE0643) (L35+36)

AAYUSH SAHAY (17BCE2141) (L35+36)

ANIKET GUPTA (17BCE2018) (L35+36)

Prepared For

DATABASE SYSTEMS (CSE2004) – PROJECT COMPONENT

Submitted To

Dr.Ramanathan L

Assistant Professor (SG)

School of Computer Science and Engineering

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Declaration

We all group members hereby declare that the project entitled “ONLINE COMMUNITY MANAGEMENT SYSTEM” is the product of our own work and has been submitted without any kind of malpractices or plagiarism.

Table of Contents

1. ABSTRACT

With the urbanization of the population as well as advancement in the technologies, there is an increasing focus on the social services. Communities have become the young generation's basic living unit. Thus, online community management system is imperative. To achieve a community online system for information management, asset management, Online Repair service, Charges inquiries and other major functionalities, through dynamic web pages to interact with the database technology. The database system is user interactive, good capability and strong security amenities. Online community management helps in connecting a large part of social services, connecting thousands of families together, but because of lack of inputs, making of such a community network is very less focused. There is a vast unexplored area in the field of online community management system. Implementation of such a system is a basic need of this generation.

2. INTRODUCTION

In the early 2000's, the concept of online community started gaining popularity. People from all around the world started participating in such communities and forming more different communities. Communities for online discussion, review, social media and other variations started to form. Quora, Stackoverflow are some of the leading online discussion-based communities nowadays. More and more communities are being formed every day. Looking at the timeline of online community development phase- first came group chat, then came email discussion form. Then came BBS. After that in recent years came, web-based communities hosted on websites and the newest is the social networking sites. As the community grew, the need for storage of data and its manipulation also grew. More advance ways of storing and data management softwares came. Those softwares ease the flow, storage and manipulation of data easily.

Online Community Management System is the process of management of data of the users, managing the data of the articles requested and posted for each community. The community acts as an interactive group of people that are able to exchange information benefitting all the users of the community. As users find answers for themselves and connect with peers for their articles, their user experience improves. Not only are users able to get more out of each other through their articles, they forge connections with others who can assist as and when any questions arise in the future in the form of articles requested leading to stronger engagement with other users. This way it becomes an interactive session for users empowering our community members.

3. PROJECT SCOPE

Online Community Management System aims at automation of the following processes

a. Community registration by users in any one of the communities of their interest
b. Users can post articles about topics related to their respective communities
c. Users can request articles from other users of the same community to gain information on topics related to the community in which they are registered in
d. Users can ask questions related to any article and answers can also be provided to these questions by other users thus engaging in an interactive environment

4. KEY CONTACTS AND SHAREHOLDERS

Name	Registration Number	Phone number
Aayush Sahay	17BCE2141	8870788836
Ashwin Goel	17BCE2073	9159733622
Amitesh Srivastava	17BCE0673	7004176249
Aniket Gupta	17BCE2018	8870799939

5. PROJECT SOURCE REQUIREMENTS

5.1 Software Requirements

Node.js

MongoDB

Express.js

5.2 Hardware Requirements

A computer system with

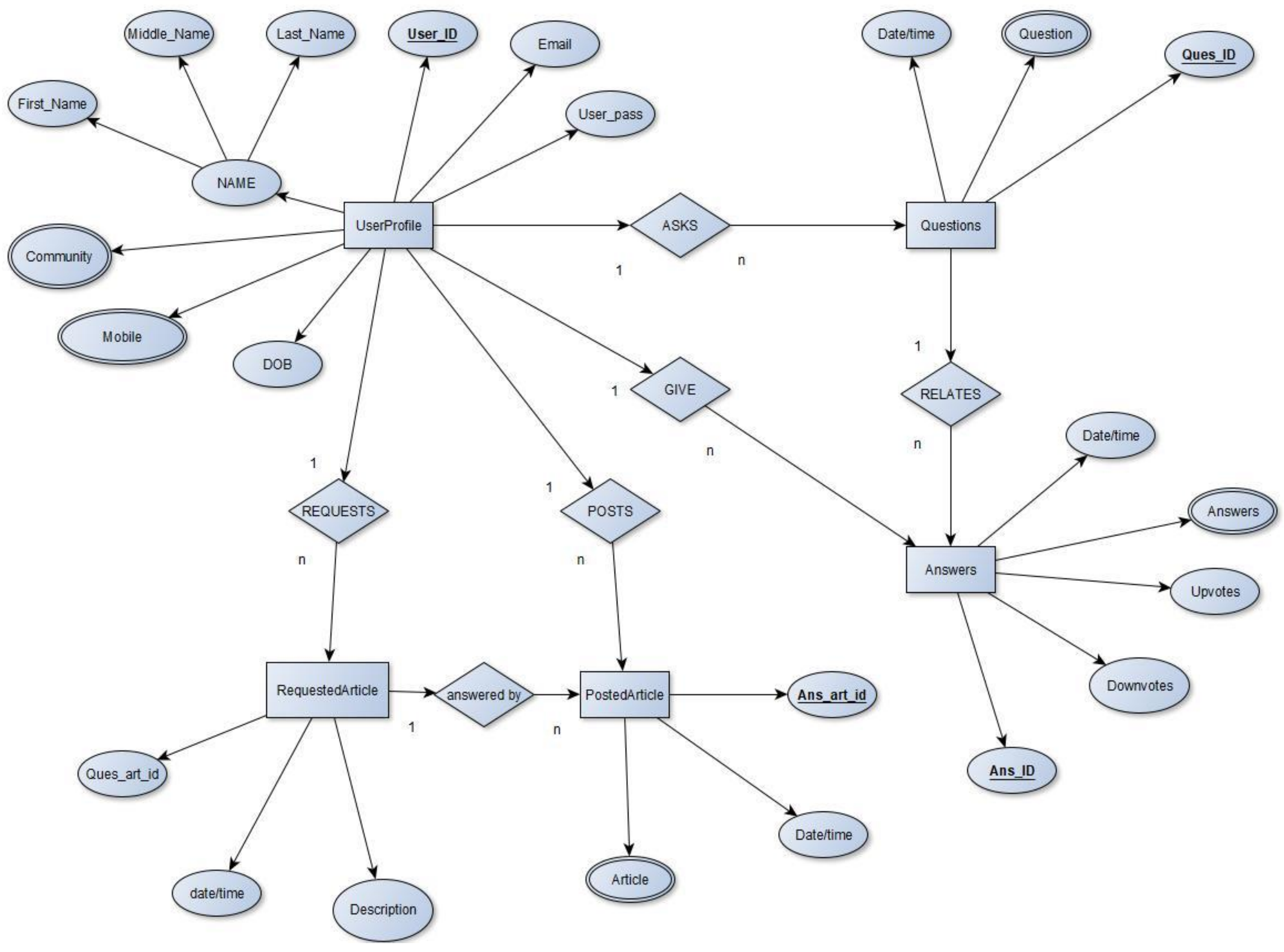
Windows -7 or above

Min 2gb ram

Intel core i5 processor

Disk space :2.5gb

6. E-R DIAGRAM



7. TABLES AND CONSTRAINTS

Table Name: UserProfile

Attribute	**Datatype**	**Constraint**
User_ID	string	Primary Key
User_pass	string	Length>6
First_Name	string	Not null
Middle_Name	string	-
Last_Name	string	Not null
Mobile	number	Not null, length=10
Email	string	Not null, unique
DOB	Date	Not null

Table Name: Questions

Attribute	Datatype	Constraint
User_ID	string	Not null, Foreign key
Ques_ID	string	Primary Key
Question	string	Not null
Date/Time	Date	Not null

Table Name: Answers

Attribute	Datatype	Constraint
User_ID	string	Not null, Foreign key
Ans_ID	string	Primary key
Ques_ID	string	Not null, Foreign key
Answer	string	Not null
Upvotes	number	-
Downvotes	number	-
Date/Time	date	Not null

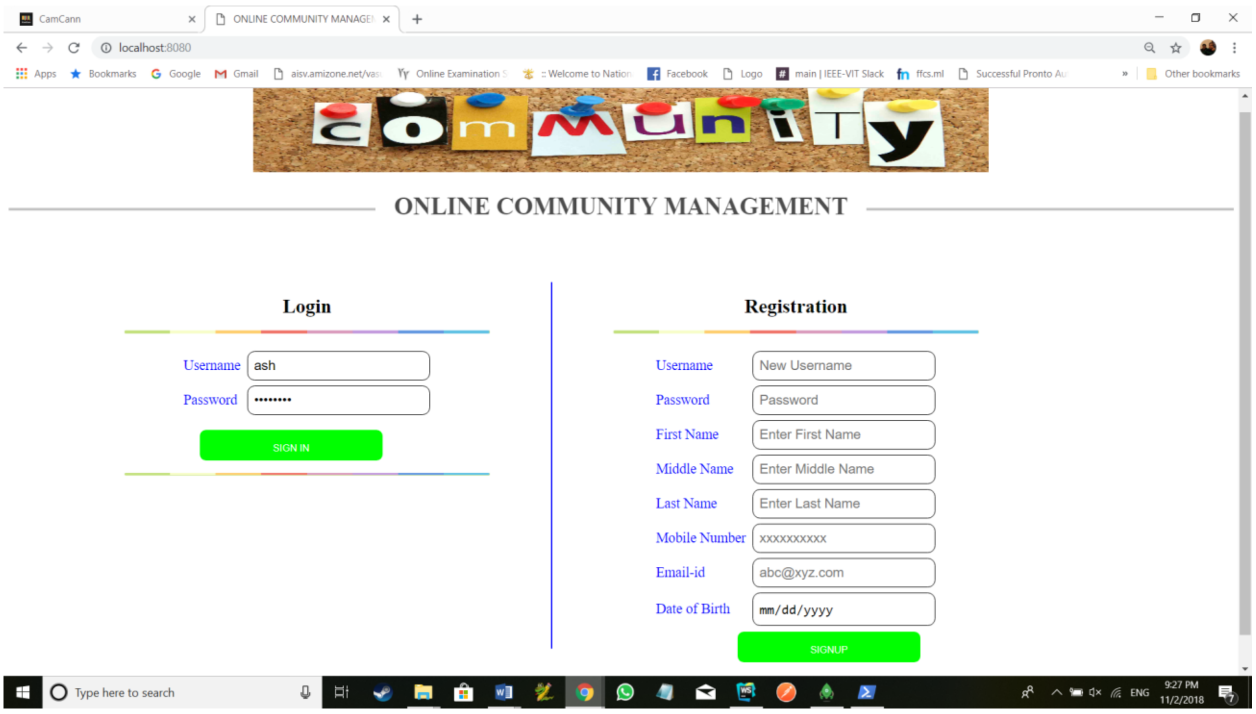
Table Name: PostedArticles

Attribute	Datatype	Constraint
User_ID	string	Not null, Foreign key
Ans_art_ID	string	Primary key
Article	string	
Date/Time	Date	Not null

Table Name: RequestedArticles

Attribute	**Datatype**	**Constraint**
User_ID	string	Not null, Foreign key
Ques_art_ID	string	Primary Key
Description	string	Length>10
Date/Time	Date	Not null

8. IMPLEMENTATION



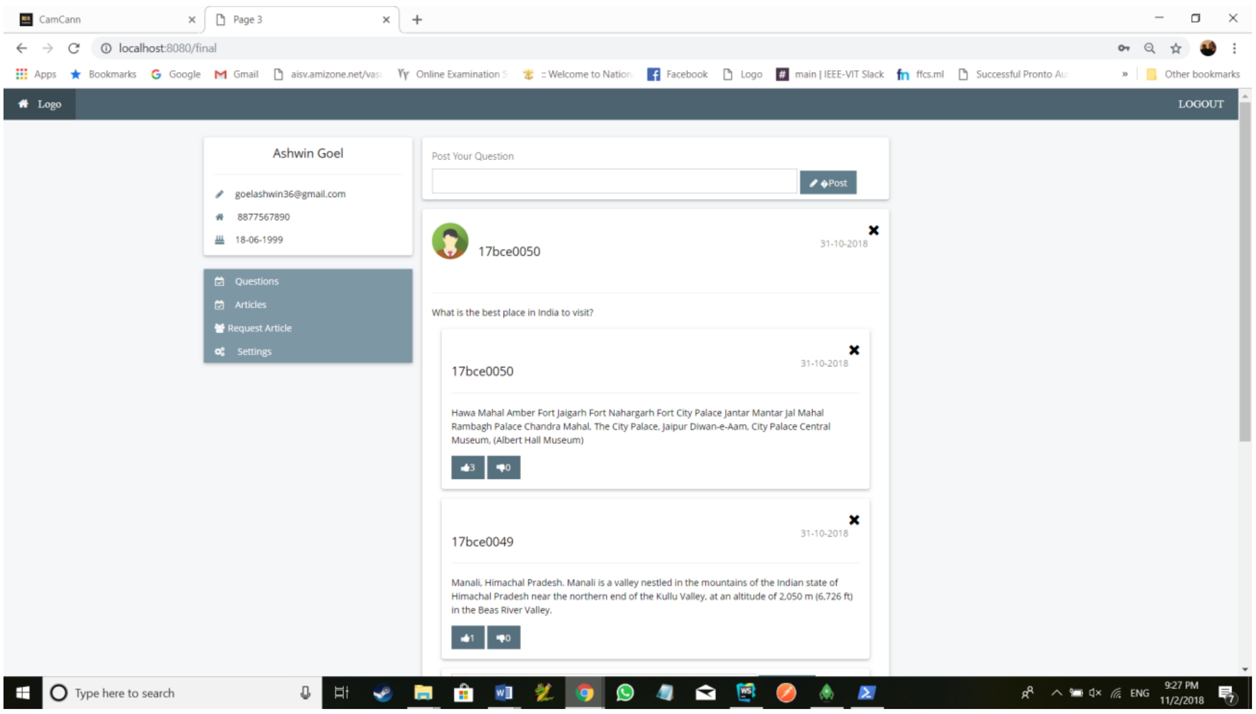
ONLINE COMMUNITY MANAGEMENT

Login

Username: ash
Password: ••••••••

Registration

Username: New Username
Password: Password
First Name: Enter First Name
Middle Name: Enter Middle Name
Last Name: Enter Last Name
Mobile Number: xxxxxxxxxx
Email-id: abc@xyz.com
Date of Birth: mm/dd/yyyy



Ashwin Goel
goelashwin36@gmail.com
8877567890
18-06-1999

Post Your Question

17bce0050 31-10-2018

What is the best place in India to visit?

17bce0050 31-10-2018

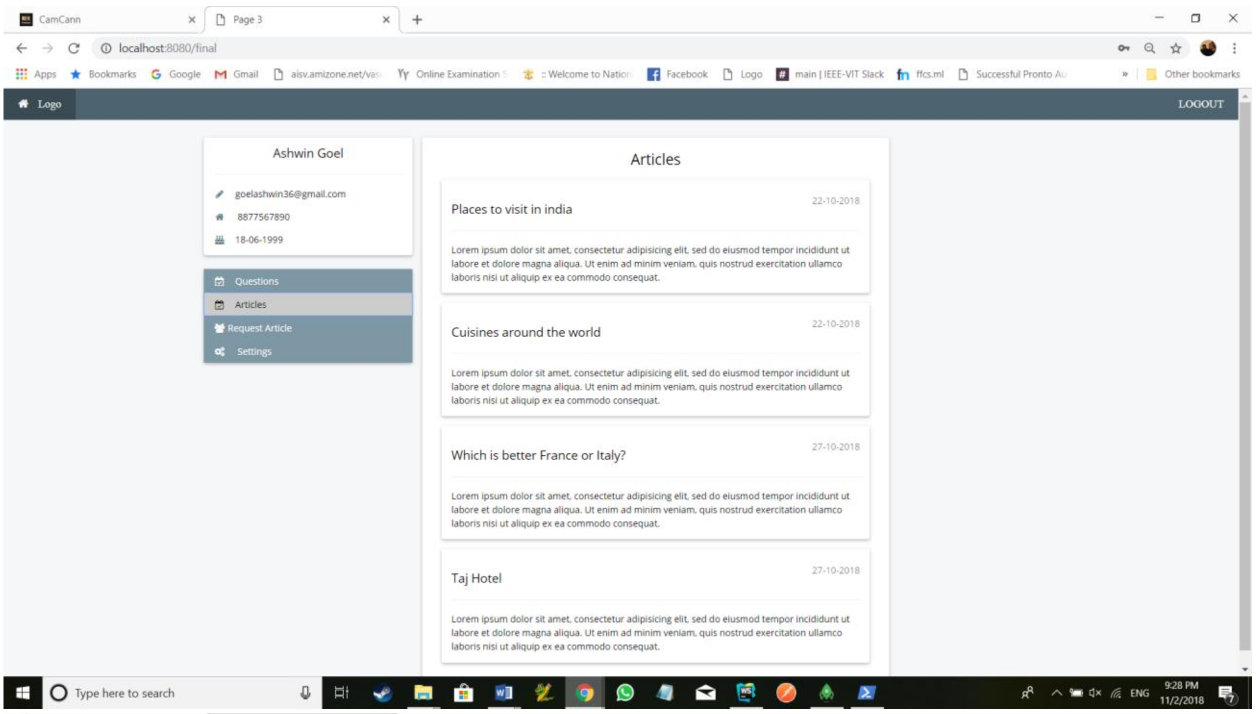
Hawa Mahal Amber Fort Jaigarh Fort Nahargarh Fort City Palace Jantar Mantar Jal Mahal Rambagh Palace Chandra Mahal, The City Palace, Jaipur Diwan-e-Aam, City Palace Central Museum, (Albert Hall Museum)

3 0

17bce0049 31-10-2018

Manali, Himachal Pradesh. Manali is a valley nestled in the mountains of the Indian state of Himachal Pradesh near the northern end of the Kullu Valley, at an altitude of 2,050 m (6,726 ft) in the Beas River Valley.

1 0



Ashwin Goel

goelashwin36@gmail.com

8877567890

18-06-1999

Articles

Places to visit in india — 22-10-2018

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Cuisines around the world — 22-10-2018

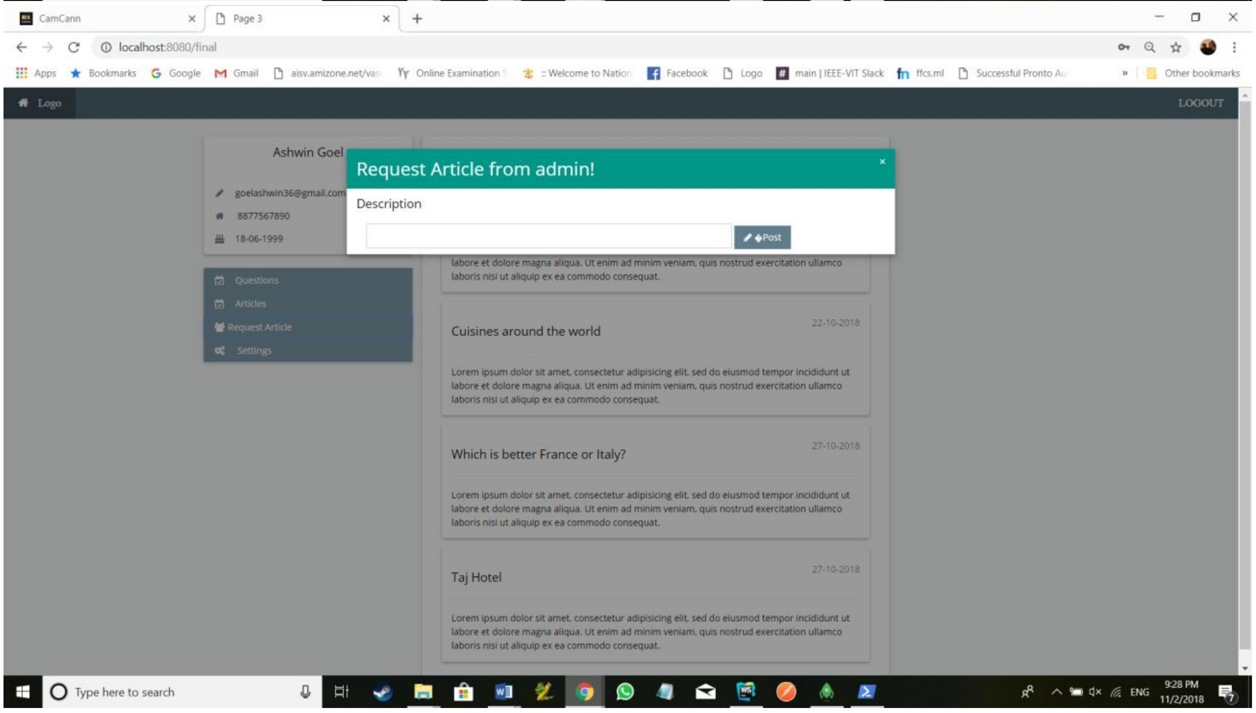
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Which is better France or Italy? — 27-10-2018

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

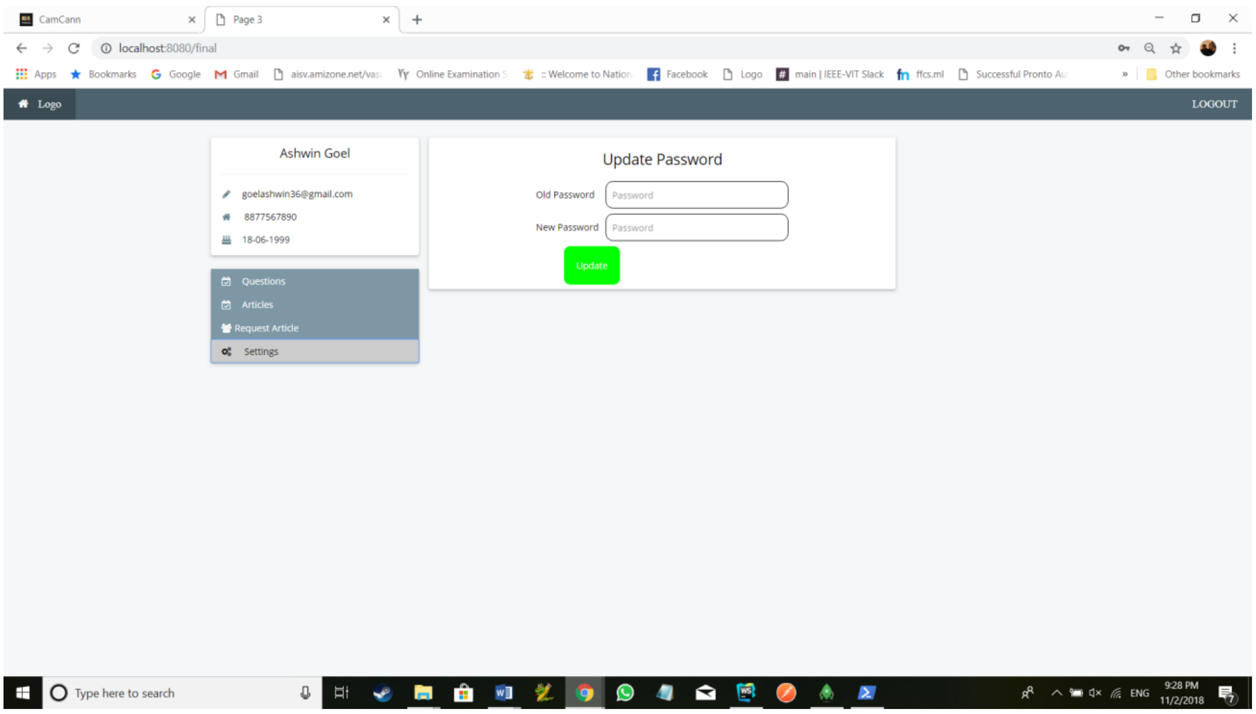
Taj Hotel — 27-10-2018

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.



Request Article from admin!

Description



SOME OF THE CODES:

Server.js

```
//Server.js

var express = require('express');
var cookieParser = require('cookie-parser');
var port = process.env.PORT || 8080;
var morgan = require('morgan');
var mongoose = require('mongoose');
var bodyParser = require('body-parser');
var path = require('path');
var getRoutes = require('./app/routes/getRoutes');
var postRoutes = require('./app/routes/postRoutes');

var app = express();

app.set('views', path.join(__dirname, 'views'));
app.set('view engine', 'ejs');

app.use(cookieParser());
app.use(bodyParser.json()); // for parsing application/json
app.use(bodyParser.urlencoded({ extended: true })); // for parsing application/x-
www-form-urlencoded
app.use(morgan('dev'));
app.use(express.static(path.join(__dirname, 'public')));

app.use(getRoutes);
app.use(postRoutes);

mongoose.connect('mongodb://localhost:27017/UserProfile', function(err){
    if(err){
        console.log("Database not connected: "+ err);
    }
    else{
        console.log("Successfully connected to the database");
    }
});


app.listen(port, function(){
    console.log("Server is running on port : " + port);
});
```

postRoutes

```
const express = require("express");
const router = express.Router();
var cookieParser = require('cookie-parser');

var User = require('../models/user');
var Ques = require('../models/ques');
var Posted_art = require('../models/posted_articles');
var Req_art = require('../models/req_articles');

router.post("/signup", function (req, res) {
    var {user_id, user_pass, email, first_name, middle_name, last_name, mobile,
dob} = req.body;
    var user = {
        user_id, user_pass, email, first_name, middle_name, last_name, mobile, dob
    };
    console.log(user);
    new User(user).save((err, docs) => {
        if (err) {
            res.send("User already exists");
            console.log(err);
            console.log(docs);
        }
        else {
            res.send("User created!");
        }
    })
});

router.post('/login', function (req, res) {
    var {user_id, user_pass} = req.body;
    console.log("\n\n" + user_id + "\n\n");
    User.find({user_id: user_id}).exec(function(err, docs){
        // res.send("err: \n" + err + "\n\n" + "docs: \n" + docs);
        console.log(req.body);
        console.log(docs);
        console.log(docs[0].user_pass);
        console.log(user_pass);
        if(err){
            res.send("Some error occurred");
            console.log(err);
        }
        else if(docs.length===0){
            res.send("User not found");
        }
        else{
            if(docs[0].user_pass == user_pass) {
                console.log(docs);
                res.cookie("login_id", user_id + "," + user_pass + "," +
docs[0]._id).send("Ok");
            }
            else if(docs[0].user_pass != user_pass){
                res.send("Invalid Password")
            }
            }
    });

});

router.post('/requestArticle', function (req, res) {
    var {user_id, description} = req.body;
    var article = {
        user_id, description
    };
    new Req_art(article).save((err) => {
        if (err) {
            res.send("Error");
            console.log(err);
        }
        else {
            res.send("Article requested.");
        }
    })
});

router.post('/post_question', function (req, res) {
    var {user_id, question, date, answers} = req.body;
    var newQuestion = {
        user_id: user_id,
        question: question,
        date: date
    };
    console.log(newQuestion);
    console.log(req.body);
    new Ques(newQuestion).save((err, docs) => {
        if (err) {
            res.send("Error");
            console.log(err);
            console.log(docs);
        }
        else {

            res.send(docs);
        }
    })
});


router.post('/post_comment', function (req, res) {
    var {user_id, ques_id, answer, upVotes, downVotes, date} = req.body;
    console.log(req.body);
    var ObjectID = require('mongodb').ObjectID;
    Ques.findById(ques_id, function(err, docs){
        console.log(docs);
    })

    Ques.findByIdAndUpdate(ques_id, {$push: {answers: {_id: new ObjectID(),
user_id: user_id, answer: answer, date: date, upVotes: [], downVotes: []}}},
function(err, response){
        if(err){
            res.send("Error");
            console.log(err);

        }
        else{
            Ques.findById(ques_id, function(err, docs){
                res.send(docs);
            })
        }
    });
});


router.post('/post_article', function(req, res){
    var {topic, article, date} = req.body;

    var newArticle = {
        topic, article, date
    };
    Posted_art(newArticle).save(function(err, docs){
       if(err){
           res.send("Error occurred");
           console.log(err);
       }
       else{
           res.send("Article created");
       }
    });
});


//-----------------------------------------------EDIT
router.post('/upVotes', function(req, res){
    var {ques_id, ans_id, user_id} = req.body;
    var y;
    console.log(req.body);

    Ques.findById(ques_id, function(err, docs) {
        if (err){
            console.log(err);
            res.send("Error");
        }
            else{
                console.log("\n\n Docs: "+docs);
                console.log("docs.ans.length: ", docs.answers.length);
               for(i=0; i<docs.answers.length; i++){
                   console.log("1");
                   console.log(docs.answers[i]._id, ans_id);
                  if(docs.answers[i]._id == ans_id){
                      console.log("\n\n2\n\n");
                      y=i;
                var u = docs.answers[y].upVotes;
                var d = docs.answers[y].downVotes;
                console.log("\n\nu: "+u);
                console.log("\n\nd: "+d);
                console.log("d.indexOf(user_id): ",d.indexOf(user_id));
                console.log("u.indexOf(user_id): ",u.indexOf(user_id));
                if(d.indexOf(user_id) !== -1){
                    console.log("\n\n3\n\n");
                    docs.answers[y].downVotes.splice(u.indexOf(user_id),1);
                    docs.answers[y].upVotes.push(user_id);
                    console.log("\n\nu: "+docs.answers[y].upVotes);
                    console.log("\n\nd: "+docs.answers[y].downVotes);
                    docs.markModified("answers");
                    docs.save(function(err, newDocs){
                        console.log("err: ",err);
                        console.log("newDocs- "+ newDocs.answers[y]);
                        res.send("up-"+(newDocs.answers[y].upVotes.length)+"-down-
"+(newDocs.answers[y].downVotes.length));
                    });
                }
                else if(u.indexOf(user_id) === -1){
                    console.log("\n\n4\n\n");

                    docs.answers[y].upVotes.push(user_id);
                    console.log("\n\nu: "+docs.answers[y].upVotes);
                    console.log("\n\nd: "+docs.answers[y].downVotes);
                    docs.markModified("answers");

                    docs.save(function(err, newDocs){

                        console.log(err);
                        console.log("newDocs- "+ newDocs);
                        console.log("newDocs.answers[0].up.length- "+
newDocs.answers[0].upVotes.length);
                        console.log("y: ",y);

                        res.send("up-"+(newDocs.answers[y].upVotes.length)+"-down-
"+(newDocs.answers[y].downVotes.length));
                       // console.log("up-"+(newDocs.answers[y].upVotes.length-
1)+"-down-"+(newDocs.answers[y].downVotes.length-1));

                    });
                }
                else{
                    console.log("\n\n5\n\n");

                    docs.answers[y].upVotes.splice(d.indexOf(user_id),1);
                    console.log("\n\nu: "+docs.answers[y].upVotes);
                    console.log("\n\nd: "+docs.answers[y].downVotes);
                    docs.markModified("answers");
                    docs.save(function(err, newDocs){
                        console.log(err);
                        console.log("newDocs- "+ newDocs.answers[y]);

                        res.send("up-"+(newDocs.answers[y].upVotes.length)+"-down-
"+(newDocs.answers[y].downVotes.length));
                        //console.log("up-"+(newDocs.answers[y].upVotes.length-
1)+"-down-"+(newDocs.answers[y].downVotes.length-1));

                    });
                }}
        }}


    });

});

router.post('/downVotes', function(req, res){
    var {ques_id, ans_id, user_id} = req.body;
    var y=0;
    console.log(req.body);

    Ques.findById(ques_id, function(err, docs) {
        if (err){
            res.send("Error");
        }
        else{
            console.log("\n\n Docs: "+docs);
            console.log("docs.ans.length: ", docs.answers.length);
            for(i=0; i<docs.answers.length; i++){
                console.log("1");
                console.log(docs.answers[i]._id, ans_id);

                if(docs.answers[i]._id == ans_id){
                    y=i;
                    var u = docs.answers[y].upVotes;
                    var d = docs.answers[y].downVotes;
                    console.log("\n\n"+u);
                    console.log("\n\n"+d);
                    console.log("d.indexOf(user_id): ",d.indexOf(user_id));
                    console.log("u.indexOf(user_id): ",u.indexOf(user_id));

                    if(u.indexOf(user_id) !== -1){
                        console.log("\n\n3\n\n");

                        docs.answers[y].upVotes.splice(d.indexOf(user_id),1);
                        docs.answers[y].downVotes.push(user_id);
                        console.log("\n\nup: "+docs.answers[y].upVotes);
                        console.log("\n\ndn: "+docs.answers[y].downVotes);
                        docs.markModified("answers");
                        docs.save(function(err, newDocs){
                            console.log("err: ",err);
                            console.log("newDocs- "+ newDocs.answers[y]);
                            res.send("up-"+(newDocs.answers[y].upVotes.length)+"-
down-"+(newDocs.answers[y].downVotes.length));
                        });
                    }
                    else if(d.indexOf(user_id) === -1){
                        console.log("\n\n4\n\n");

                        docs.answers[y].downVotes.push(user_id);
                        console.log("\n\nup: "+docs.answers[y].upVotes);
                        console.log("\n\ndn: "+docs.answers[y].downVotes);
                        docs.markModified("answers");
                        docs.save(function(err, newDocs){
                            console.log("newDocs- "+ newDocs);
                            console.log("y: ",y);

                            res.send("up-"+(newDocs.answers[y].upVotes.length)+"-
down-"+(newDocs.answers[y].downVotes.length));
                        });
                    }
                    else{
                        console.log("\n\n5\n\n");

                        docs.answers[y].downVotes.splice(u.indexOf(user_id),1);
                        console.log("\n\nup: "+docs.answers[y].upVotes);
                        console.log("\n\ndn: "+docs.answers[y].downVotes);
                        docs.markModified("answers");
                        docs.save(function(err, newDocs){
                            console.log(err);
                            console.log("newDocs- "+ newDocs.answers[y]);
                            res.send("up-"+(newDocs.answers[y].upVotes.length)+"-
down-"+(newDocs.answers[y].downVotes.length));
                        });
                        break;
                    }}
            }}
    });
});


router.post('/delete_question', function (req, res) {
    var {ques_id} = req.body;
    var user_id = req.cookies.login_id.split(',')[0];
    console.log(ques_id, user_id);
    Ques.findById(ques_id, function(err, docs){
        if(docs.user_id === user_id){
            Ques.deleteOne({_id: ques_id}, function(err, newDocs){
                console.log(newDocs);
                if(err){
                    res.send("Error Occurred");
                }
                else{
                    res.send("Question deleted");
                }
            })
        }
    });

});

router.post('/delete_comment', function (req, res) {
    var {ques_id, ans_id} = req.body;
     var user_id = req.cookies.login_id.split(',')[0];
     console.log("User_id: ", user_id);
    console.log(ques_id, user_id);
    Ques.findById(ques_id, function(err, docs){
        console.log(docs);
        for(i=0; i<docs.answers.length; i++){
            var ans = docs.answers[i];
            console.log("ans.user_ix: ", ans.user_id);

            if (docs.answers[i].user_id === user_id){
                var y = i;
                console.log("Same");
                docs.answers.splice(y,1);
                docs.save(function(err, newDocs){
                    if(err){
                        console.log(err);
                        res.send("Error");
                    }
                    else{
                        res.send("Comment Deleted");
                    }
                });
            }
        }
    });
});


router.post('/updatePass', function(req, res){
    var {user_id, old_pass, new_pass} = req.body;
    console.log(req.body);
    User.findById(user_id, function(err, docs){
        console.log("docs: ",docs);
        console.log("docs.user_pass: ",docs.user_pass == old_pass);

        if(err){
            console.log(err);
        }
        else if(docs.user_pass == old_pass){
            docs.user_pass = new_pass;
            console.log("new docs: ", docs);
            docs.save(function(err, newDocs){
                if(err){
                    console.log(err);
                    res.send("Error");
                }
                else{
                  //  res.clearCookie('login_id', {path:
'/'}).status(200).send('Ok.');
                    res.cookie("login_id",
newDocs.user_id+","+newDocs.user_pass+","+newDocs._id).send("Password Successfully
Updated");
                    console.log("Cookie set");
                }
            })
        }
        else{
            res.send("Wrong Password");
        }
    });

    });


module.exports = router;
```

ajax calls to server example

```
$('#signup').click(function(){
    var username = $("input[name='user_reg']").val();
    var password = $("input[name='pass_reg']").val();
    var first = $("input[name='first']").val();
    var mid = $("input[name='mid']").val();
    var last = $("input[name='last']").val();
    var mob = $("input[name='mob']").val();
    var email = $("input[name='email']").val();
    var bday = $("input[name='bday']").val();

    console.log(username+"\n"+password);

    $.ajax(
        {
            type: "POST",
            url: "http://localhost:8080/signup",
            data: {
                "user_id": username,
                "user_pass": password,
                "email": email,
                "first_name": first,
                "middle_name": mid,
                "last_name": last,
                "mobile": mob,
                "dob": bday,
            },
            success: function (response) {
                alert(response);
            }
        });
    return 0;
});
```

9. WORK BREAKDOWN

Registration number	Name	Work Assigned
17BCE2141	Aayush Sahay	Frontend, Research, Documentation
17BCE2073	Ashwin Goel	Backend, Database Design
17BCE0643	Amitesh Srivastava	Frontend, Database Design
17BCE2018	Aniket Gupta	Frontend, Database Design