Dharmsinh Desai University, Nadiad

Faculty of Technology, Department of Computer Engineering

B.Tech. CE Semester – V

Subject: (CE-515) Advanced Technologies

Project Title:

Online Discussion Forum

By:

Raj Kalathiya G. (Roll No: CE055 ID: 18CEUOS132)

Guided By:

Prof. Prashant M. Jadhav

Prof. Ankit P. Vaishnav

Dharmsinh Desai University, Nadiad

Faculty of Technology, Department of Computer Engineering

CERTIFICATE

This is to certify that Advanced Technologies project entitled “Online Discussion Forum” is the bonafied report of work carried out by **Raj Gordhanbhai kalathiya (18CEUOS132)** Of Department of Computer Engineering, Semester V, academic year 2019-20, under our supervision and guidance.

Guide	Guide	Guide
Prof. Prashant Jadhav	**Prof. Ankit Vaishnav**	**Dr. C. K. Bhensdadia**
Assistant Professor of Department of Computer	Assistant Professor of Department of Computer	Head of the Department
Engineering, Dharmsinh Desai University, Nadiad.	Engineering, Dharmsinh Desai University, Nadiad.	Engineering, Dharmsinh Desai University, Nadiad.

Abstract:

The Online Discussion forum enables Users to Create a Forum and they can join Other’s forum, and a system administrator delete some forum if found any suspicious discussion. users also can delete the forum which they have created. They can view the news on the same platform. They can like or dislike some other user’s Message.

Introduction:

This site provides a platform to users where they can discuss their problems by creating a forum (one type of room). Users need to register his/her self on site After submitting the data site will display one popup in which users need to enter otp which is provided to them by mail after successfully registration users will be redirected to the login page. Users need to be logged in to enter other's forums or create their own forum. To create a forum users need to click on the create button and enter the Forum name in the input box. Users can see in which forum they joined previously and users can also leave that forum. Users can join other's forums by unique id.There is one admin who can delete those forums which are found in suspicious discussion. If user want to leave the forum then he/she can click on leave button.

It also provides users to easily change password if any user has forgotten their password. If the user has any type of query they can also contact us through the email provided by us. They can also contact us through social media platforms like instagram, facebook. They can also call us for any type of query.

Users should be familiar with login accounts, register new accounts. But if the user is not familiar then first he/she suggests to read user documentation. Full internet connection is required.

Technology Used:

- Hyper Text Markup Language(HTTP)
- Cascading Style Sheet (CSS 3) for styling the HTML pages.
- JavaScript for providing dynamic content for the HTML pages.
- Jquery to provide dynamic data to the request pages.
- Bootstrap4 for styling the HTML pages using predefined classes of bootstrap.
- Angular Framework.
- Express.js for routing purposes.
- Node.js for backend purposes.
- Mongo DB as a database manager to add, update and fetch data from the database using Node.js mongoose module.

Platform Used:

- Google Chrome
- Mozilla Firefox

Tools Used:

- I used Visual Studio Code for development of the code.
- Github as a version control method and to manage the project.

Software Requirement Specification

1. FUNCTIONAL REQUIREMENTS:

1.1 Registration ,Login and Manage user Account

Description :

If a user wants to use the site then he/she must be registered, unregistered users can't use the system.Customer logins to the system by entering valid user id and password for the Forum.

1.1.1 Register new user

Input: For registration user should provide Username, address, password, phone no, email id etc..
Output : Pop up will be shown in which user need to enter the otp after successful register Message shown that Your account has been registered and redirect to login page

1.1.2 Login user

Input : Username and password
Output: if username and password is correct then it redirect to Forum Page Otherwise Invalid details message printed.

1.1.3 Manage User Account Detail

Input: Collect edited detail from user
Output : Account Detail is Updated message

2.1 Manage Forum

Description :

If a user wants to create or join the forum then he/she must be logged in first.

2.1.1 Create new forum

Input: Enter the forum name and then click on create button

Output: Forum Created successfully

2.1.2 Join new forum

Input: Enter ForumId and click on join button

Output: If all is ok then the user will join the forum.

2.1.3 Leave from the forum

Input: click on leave button

Output: user leaves successfully

3.1 Administrator

State : Admin must Logged in his/her Account

3.1.1 Delete Forum

Input: Click on delete button

Output: After Confirmation Forum will be deleted.

3.1.2 Remove user

Input: click on block button in user’s list

Output: User has been removed.

2. Other Nonfunctional Requirements

2.1 Performance Requirements

The system shall accommodate a high number of Forums and users without any fault.Access is given to only valid users.

2.2 Safety Requirements

System use shall not cause any harm to users.

2.3 Security Requirements

System will use secured database

Normal users can just read information but they cannot edit or modify anything except their forum and some other information.

2.4 Error Handling

Online Discussion forum shall handle expected and non expected errors in ways that prevent loss in information and long downtime period.

Design Documents

1. Xml

Forums:

```
<Forums>
        <Forums id="F_01">
                <forumname>ABCXYZ</forumname>
                <email>ABC12345@gmail.com</email>
                <forumid>603, SDFG</forumid>
        </Forums>
</Forums>
```

Owner:

```
<Owners>
        <Owner id="O_01">
                <forumname>ABCXYZ</forumname>
                <forumid>603, SDFG</forumid>
                <email>ABC12345@gmail.com</email>
        </Owner>
</Owners>
```

User:

```
<users>
        <user id="U_01">
                <firstname>Raj</firstname>
                <lastname>Kalathiya</lastname>
                <email>ABC12345@gmail.com</email>
                <password>1234</password>
        </user>
</users>
```

DiscussMsg:

```
<discussMsgs>
        <discussMsg id="U_01">
                <discussid>66732red3yu2q</discussid>
                <email>ABC12345@gmail.com</email>
                <msg>1234</msg>
        </discussMsg>
</discussMsgs>
```

2. DTD

Forums:

```
<!DOCTYPE forums[
<!ELEMENTS forums(forums+)>
<!ELEMENTS forums(forumname,email,forumid)>
<!ELEMENT forumname(#PCDATA)>
<!ELEMENT email(#PCDATA)>
<!ELEMENT forumid(#PCDATA)>
```

<#ATTLIST forum id ID #required>

Owners:

<!DOCTYPE owners[

<!ELEMENTS owners(owners+)>

<!ELEMENTS owners(forumname,email,forumid)>

<!ELEMENT forumname(#PCDATA)>

<!ELEMENT email(#PCDATA)>

<!ELEMENT forumid(#PCDATA)>

<#ATTLIST forum id ID #required>

User:

<!DOCTYPE users[

<!ELEMENTS users(users+)>

<!ELEMENTS users(forumname,email,forumid)>

<!ELEMENT firstname(#PCDATA)>

<!ELEMENT lastname(#PCDATA)>

<!ELEMENT email(#PCDATA)>

<!ELEMENT password(#PCDATA)>

<#ATTLIST forum id ID #required>

DiscussMsg:

<!DOCTYPE discussmsgs[

<!ELEMENTS discussmsgs(discussmsg+)>

```
<!ELEMENTS discussmsg(discussid, firstname, lastname, email, msg)>
<!ELEMENT discussid(#PCDATA)>
<!ELEMENT email(#PCDATA)>
<!ELEMENT msg(#PCDATA)>
<#ATTLIST discussmsg id ID #required>
```

3. XSD

Forums:

```
   <xs:schema attributeFormDefault="unqualified"
elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Forums">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Forums">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="forumname" type="xs:string" />
              <xs:element name="email" type="xs:string" />
              <xs:element name="forumid" type="xs:string" />
            </xs:sequence>
            <xs:attribute name="id" type="xs:string" use="required" />
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Owners:

```
   <xs:schema attributeFormDefault="unqualified"
elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Owners">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Owner">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="forumname" type="xs:string" />
              <xs:element name="forumid" type="xs:string" />
              <xs:element name="email" type="xs:string" />
            </xs:sequence>
            <xs:attribute name="id" type="xs:string" use="required" />
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
```

```
</xs:schema>
```

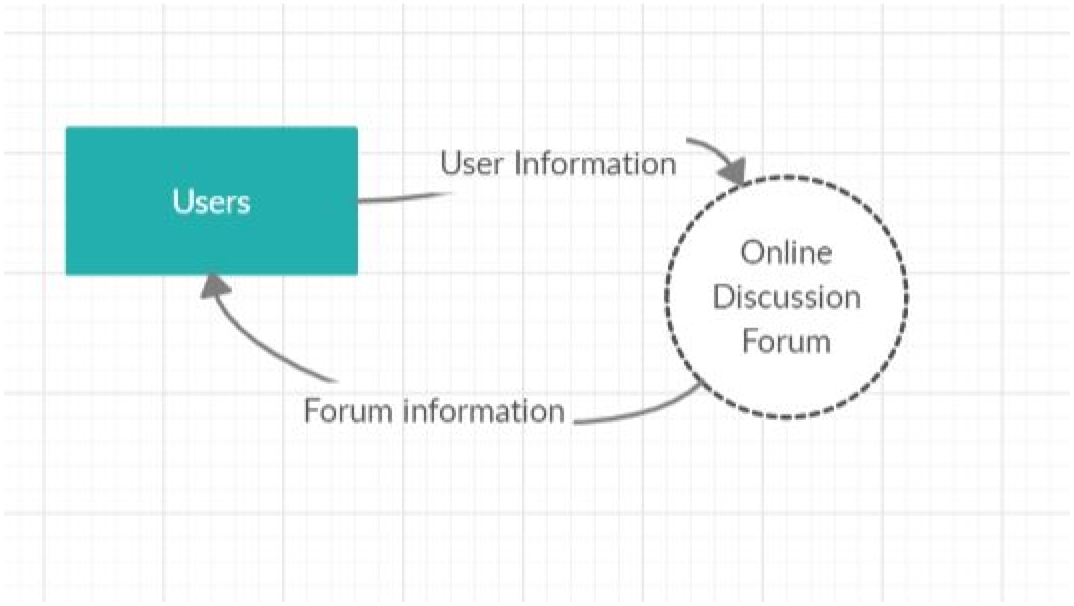
User:

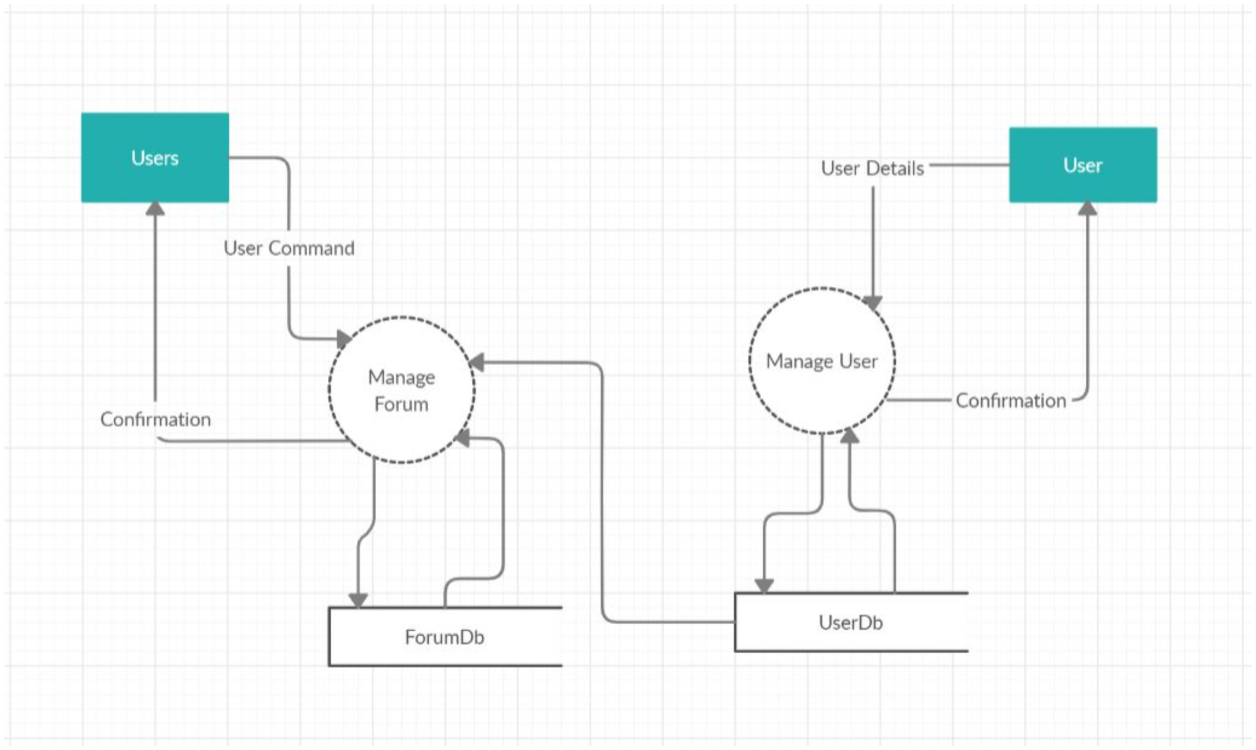
```
<xs:schema attributeFormDefault="unqualified"
elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="users">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="user">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="firstname" type="xs:string" />
              <xs:element name="lastname" type="xs:string" />
              <xs:element name="email" type="xs:string" />
              <xs:element name="password" type="xs:unsignedShort" />
            </xs:sequence>
            <xs:attribute name="id" type="xs:string" use="required" />
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

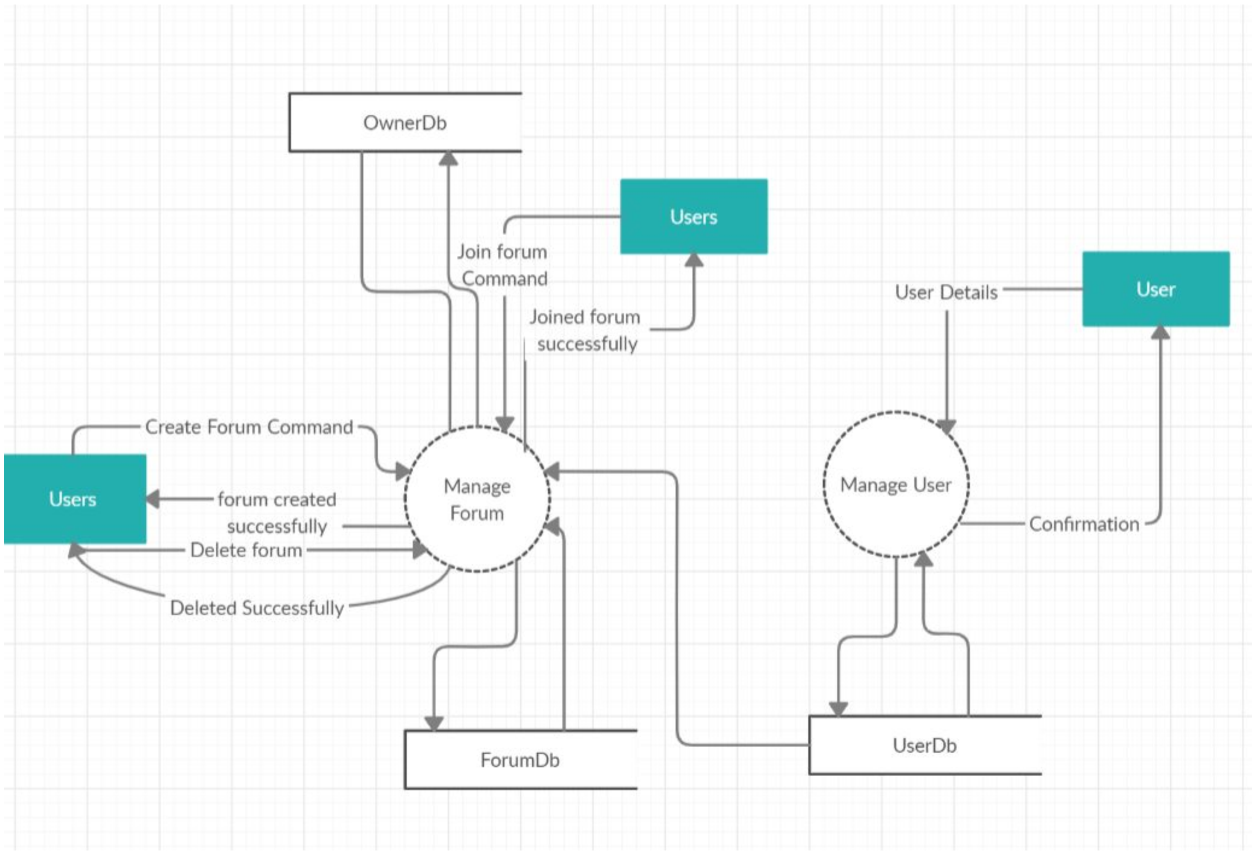
DiscussMsg:

```
<xs:schema attributeFormDefault="unqualified"
elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="discussMsgs">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="discussMsg">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="discussid" type="xs:string" />
              <xs:element name="email" type="xs:string" />
              <xs:element name="msg" type="xs:unsignedShort" />
            </xs:sequence>
            <xs:attribute name="id" type="xs:string" use="required" />
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

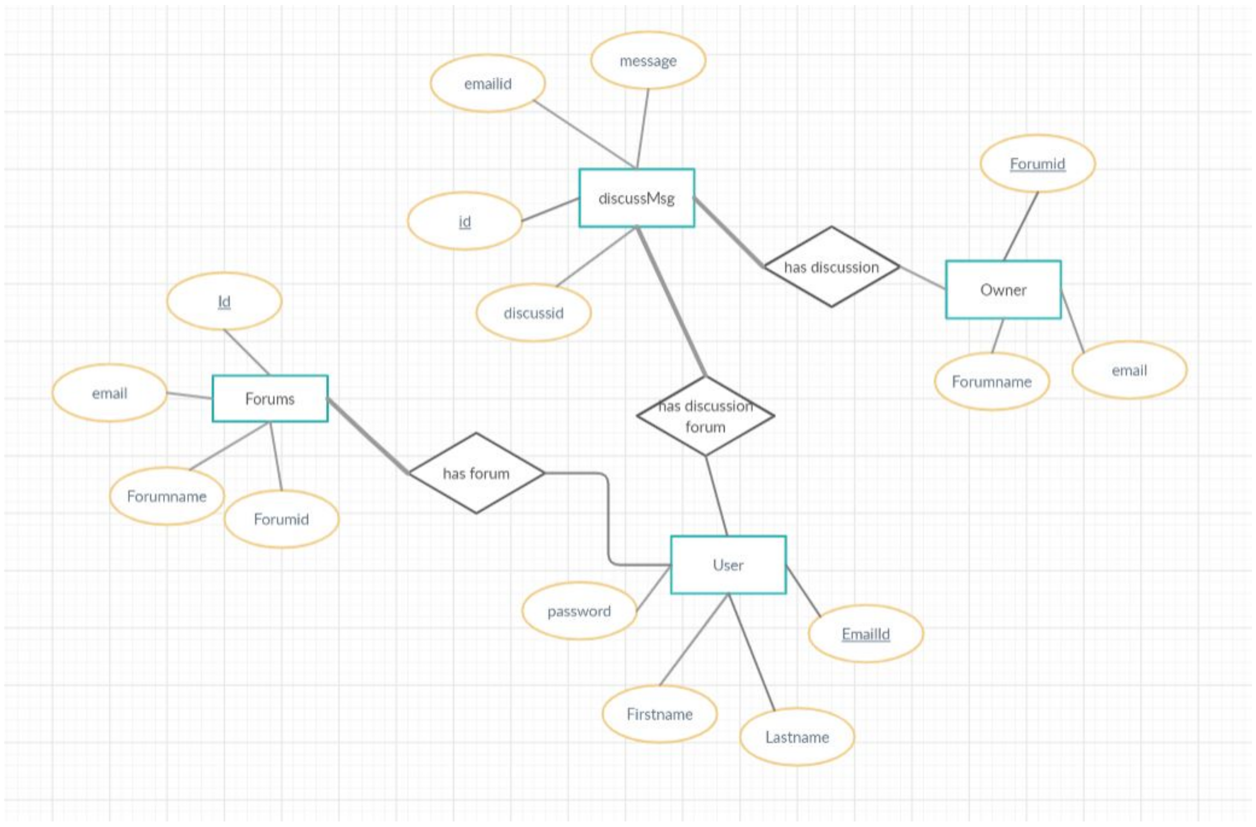
2.DFD


Users
User Information
Online Discussion Forum
Forum information


Users
User Command
Confirmation
Manage Forum
ForumDb
User Details
User
Manage User
Confirmation
UserDb


OwnerDb
Users
Join forum Command
Joined forum successfully
User Details
User
Create Forum Command
Users
forum created successfully
Delete forum
Deleted Successfully
Manage Forum
Manage User
Confirmation
ForumDb
UserDb

3. ER diagram


emailid
message
Forumid
discussMsg
id
has discussion
Owner
Id
discussid
email
Forums
Forumname
email
has discussion forum
has forum
Forumname
Forumid
User
password
EmailId
Firstname
Lastname

4.Data Dictionary

Users:

No	Field Name	Datatype	Width	Required	Unique	pk/fk	Referenced table	description
1	id	varchar	255	true	true			Auto Increment
2	firstname	varchar	255	true				
3	lastname	varchar	255	true				
4	email	varchar	255	true	true	true		
5	password	varchar	255	true				

Forums:

No	Field Name	Datatype	Width	Required	Unique	pk/fk	Referenced table	description
1	id	varchar	255	true	true	true		Auto Increment
2	forumname	varchar	255	true				
3	forumid	varchar	255	true	true	true	users	
4	email	varchar	255	true	true			

DiscussMsg:

No	Field Name	Datatype	Width	Required	Unique	pk/fk	Referenced table	description
1	id	varchar	255	true	true	true		Auto Increment
2	discussid	varchar	255	true		true	Owner	
3	email	varchar	255	true	true	true	Users	
4	msg	varchar		true				

Owners:

No	Field Name	Datatype	Width	Required	Unique	pk/fk	Referenced table	description
1	forumid	varchar	255	true	true	true		
2	forumname	varchar	255	true				
3	email	varchar	255	true	true			

implementation Details

Login Module:-

Users have to provide login details to this module to access for the authentication. If a user's email is not present in the database then they get an error message about that and on an unsuccessful attempt of login, they have to re-enter the login details.if the user forgot password then they can click on forget password after that they can change it.

Register Module:-

Users have to provide their personal details to register. If any user will enter email that is already in the database then they will get an error about that. Users and Merchants have to provide valid and strong information to continue with the website otherwise they will get error messages. After hitting the submit button the system sends a mail to the user which contains otp. That otp needs to be entered by the user to verify the user. If provided otp is valid then the account of the user will be created and the user will be redirected to the login page.

Forum Module:-

After successfully registering and logging in, users can see Forums. In that forum user has two buttons to create a new forum user needs to click on create new forum and then enter the forum name in the input box and then click on create button. To join another forum user needs to click on the join button after that he/she needs to enter the id and then click on the join button if the id is available and the user has not joined already then the user will join that forum otherwise an appropriate error message is shown. After successfully creating or joining a forum on that page, the user can see that forum.

Index Module:-

As the home page is unveiled, it starts off with a loading animation and after the loading is complete the animation of particles is visible along with navbar to go to the desired location of the current page or other page like login page or registration page by click on button.

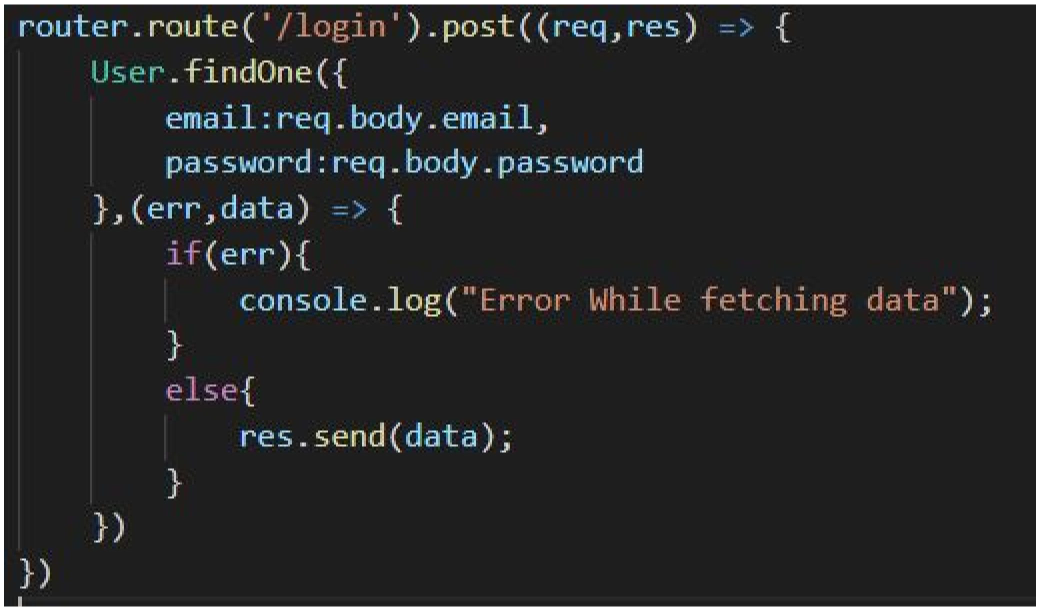
Admin Module:-

Admin can delete any forum if they find it suspicious that admin can remove the user. Admin must log in to site.

Function Prototype:

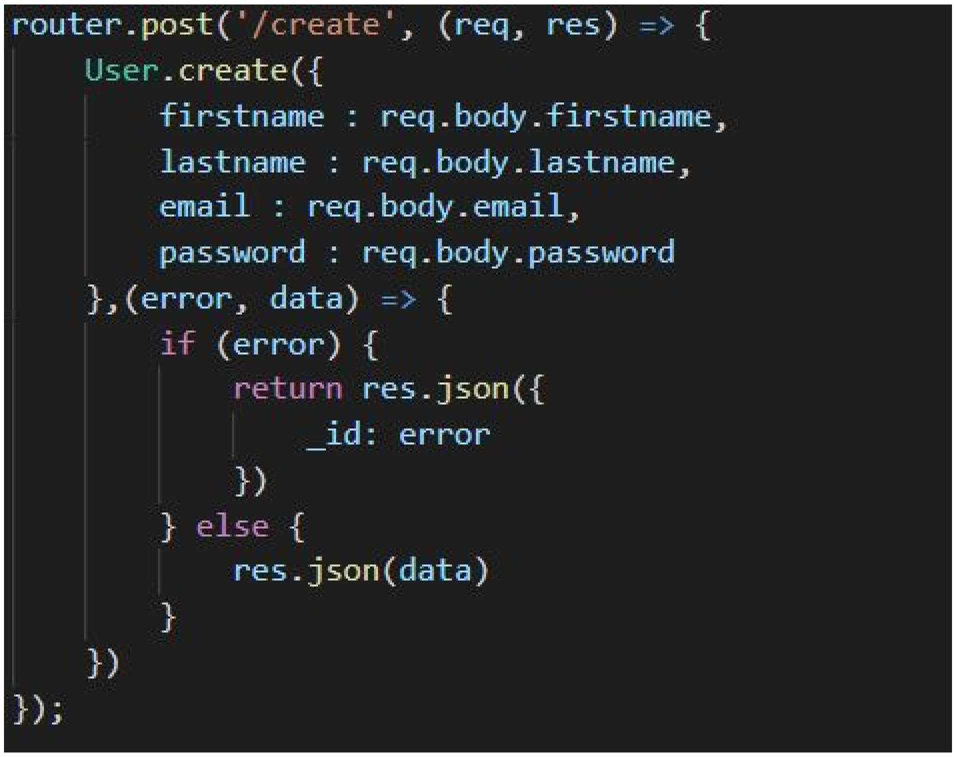
Users:

public login(data): Observable<User>



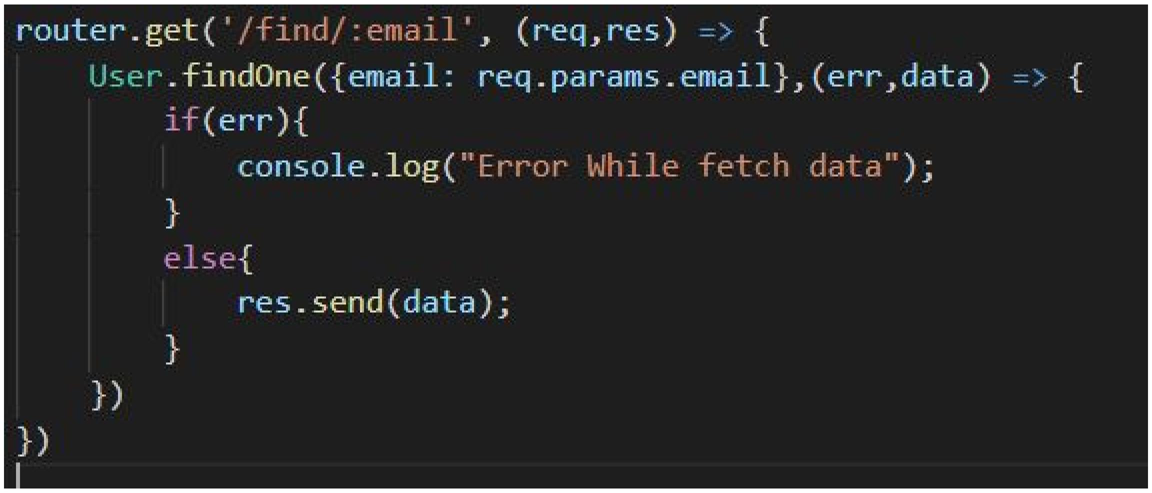
```
router.route('/login').post((req,res) => {
    User.findOne({
        email:req.body.email,
        password:req.body.password
    },(err,data) => {
        if(err){
            console.log("Error While fetching data");
        }
        else{
            res.send(data);
        }
    })
})
```

public create(data): Observable<User>



```
router.post('/create', (req, res) => {
    User.create({
        firstname : req.body.firstname,
        lastname : req.body.lastname,
        email : req.body.email,
        password : req.body.password
    },(error, data) => {
        if (error) {
            return res.json({
                _id: error
            })
        } else {
            res.json(data)
        }
    })
});
```

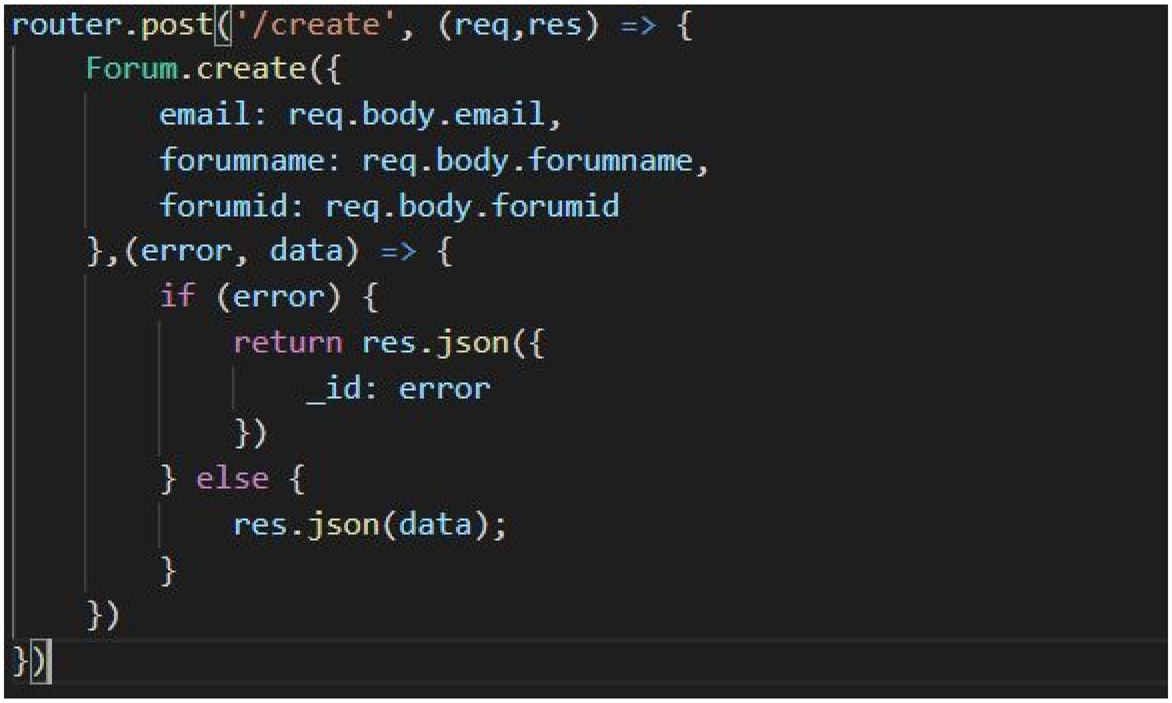
Public get(email) Observable<User>



```
router.get('/find/:email', (req,res) => {
    User.findOne({email: req.params.email},(err,data) => {
        if(err){
            console.log("Error While fetch data");
        }
        else{
            res.send(data);
        }
    })
})
```

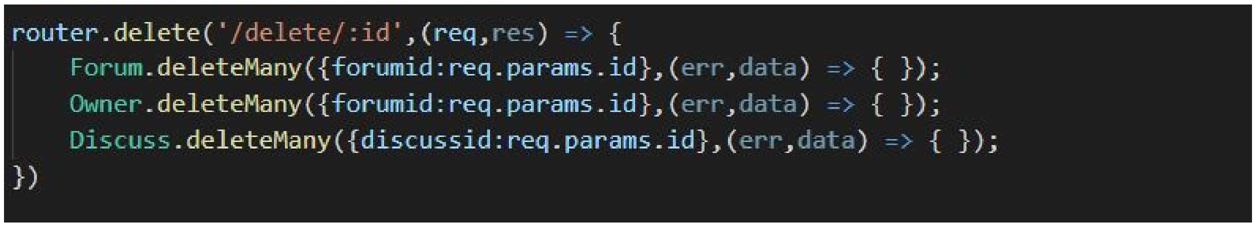
Forums:

public create(data): Observable<User> // To create a new Forum



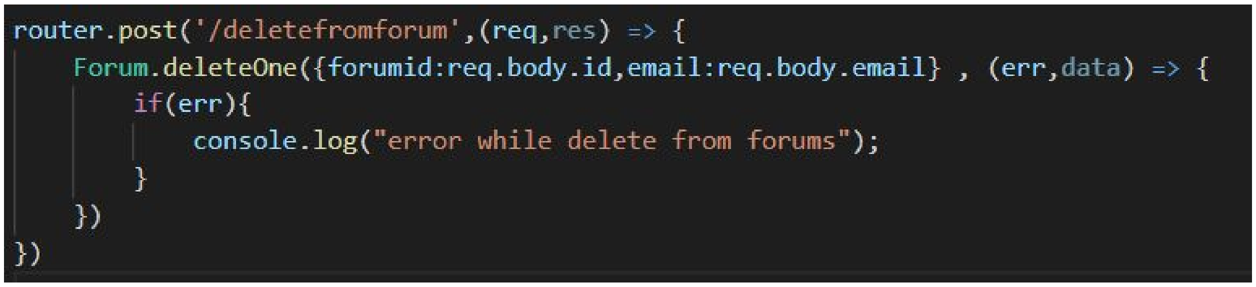
```
router.post('/create', (req,res) => {
    Forum.create({
        email: req.body.email,
        forumname: req.body.forumname,
        forumid: req.body.forumid
    },(error, data) => {
        if (error) {
            return res.json({
                _id: error
            })
        } else {
            res.json(data);
        }
    })
})
```

Public delete(id) // To delete forum only admin or owner of forum can do this



```
router.delete('/delete/:id',(req,res) => {
    Forum.deleteMany({forumid:req.params.id},(err,data) => { });
    Owner.deleteMany({forumid:req.params.id},(err,data) => { });
    Discuss.deleteMany({discussid:req.params.id},(err,data) => { });
})
```

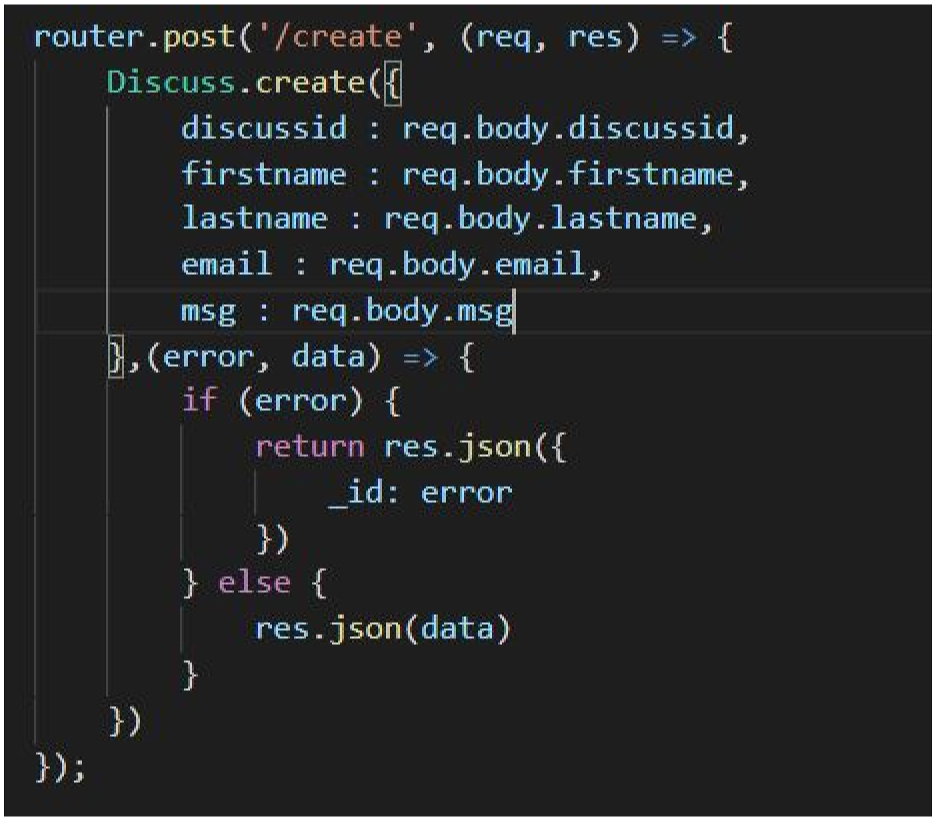
Public deletefromforum(data) //To leave from the forum



```
router.post('/deletefromforum',(req,res) => {
    Forum.deleteOne({forumid:req.body.id,email:req.body.email} , (err,data) => {
        if(err){
            console.log("error while delete from forums");
        }
    })
})
```

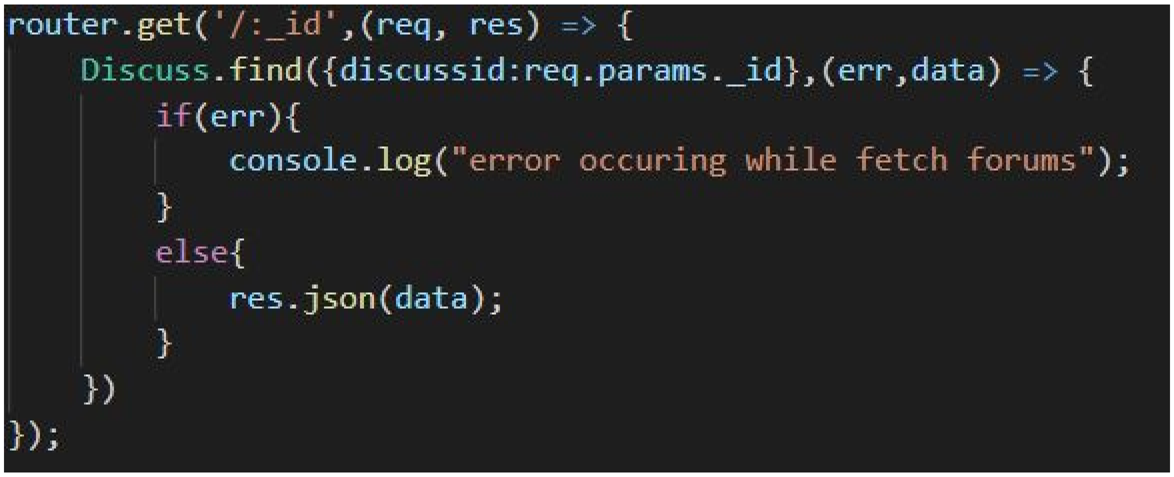
DiscussMsg:

Public create(data): Observable<Discuss> //To store the message which is sent by user



```
router.post('/create', (req, res) => {
    Discuss.create({
        discussid : req.body.discussid,
        firstname : req.body.firstname,
        lastname : req.body.lastname,
        email : req.body.email,
        msg : req.body.msg
    },(error, data) => {
        if (error) {
            return res.json({
                _id: error
            })
        } else {
            res.json(data)
        }
    })
});
```

Public get(_id): Observable<Discuss> //To get the data for forum



```
router.get('/:_id',(req, res) => {
    Discuss.find({discussid:req.params._id},(err,data) => {
        if(err){
            console.log("error occuring while fetch forums");
        }
        else{
            res.json(data);
        }
    })
});
```

Testing:

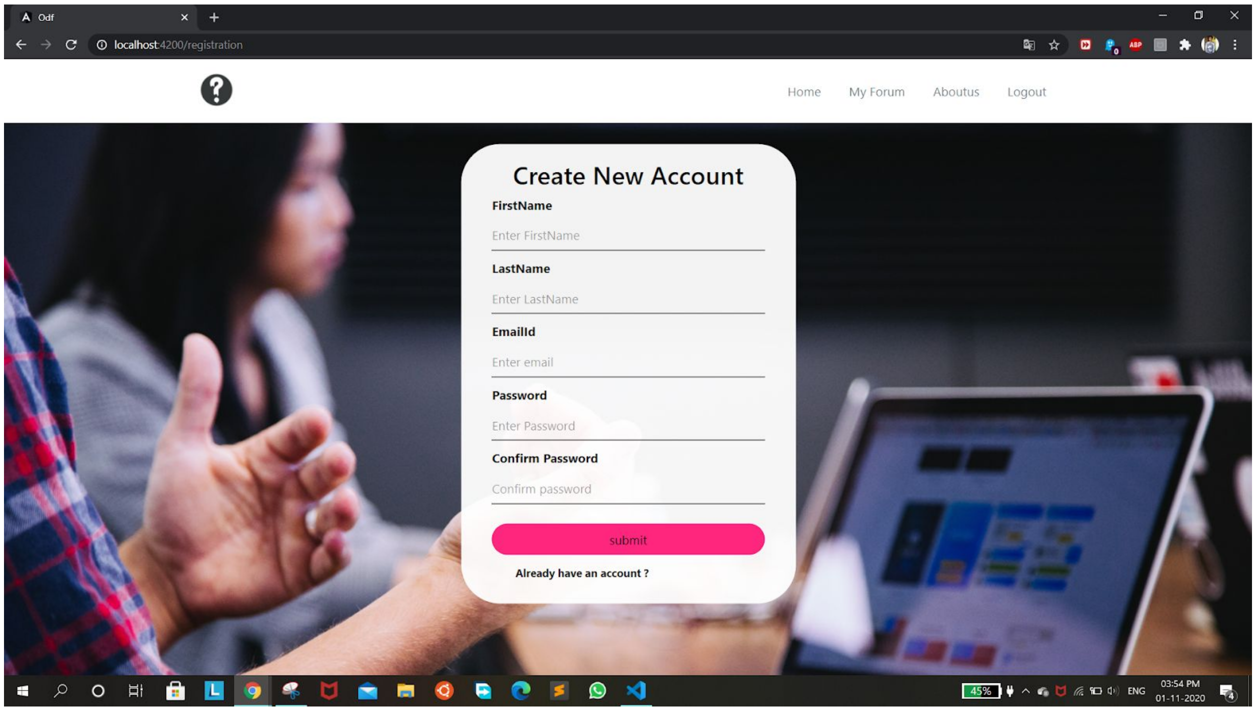
Testing for the backend API is done with the help of postman and the testing method used is unit testing. e.g.

```
{
“firstname”: "Raj"
“lastname”: "Kalathiya"
“email”: "rajkalathiya699@gmail.com"
“password”:"Raj@123”
}
```

```
{
“Discussid”:”1234”
“Firstname”:”raj”
“lastname”:”kalathiya”
“emai;”:”rajkalathiya699@gmail.com”
“msg”:”Hey this is At project”
}
```

Testing for fronted is done with manual testing.

ScreenShots



Create New Account

FirstName

Enter FirstName

LastName

Enter LastName

EmailId

Enter email

Password

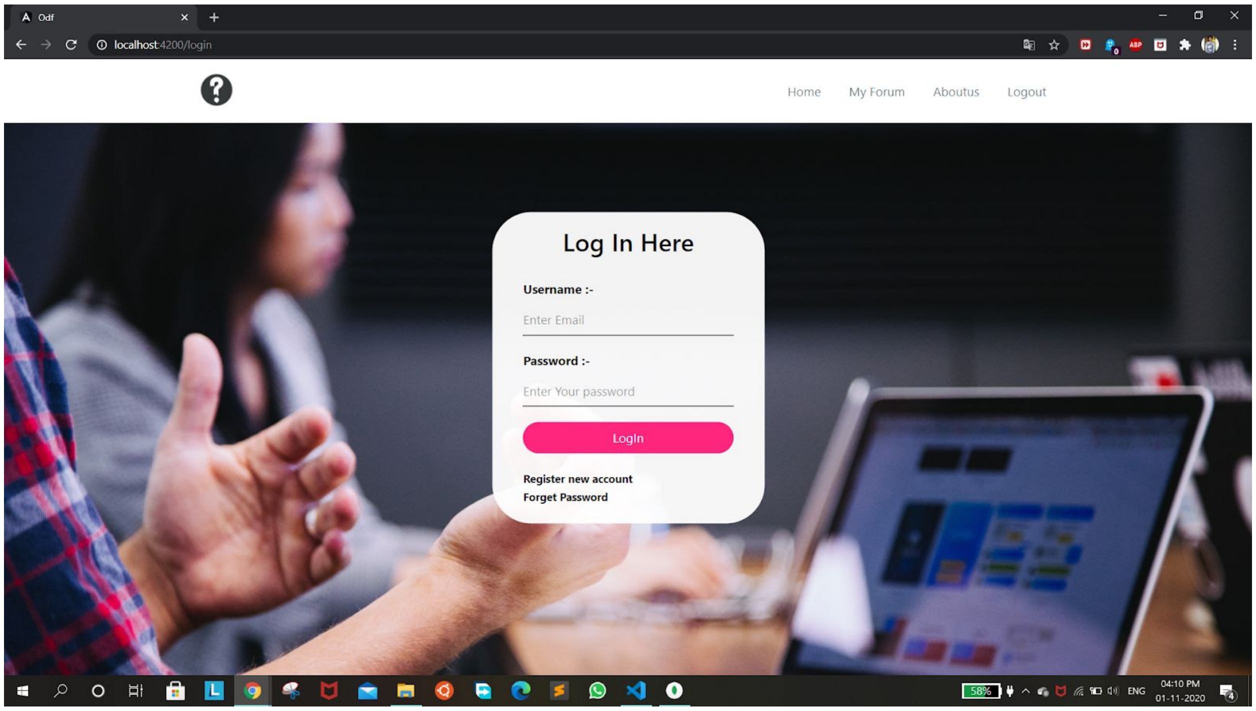
Enter Password

Confirm Password

Confirm password

submit

Already have an account ?



Log In Here

Username :-

Enter Email

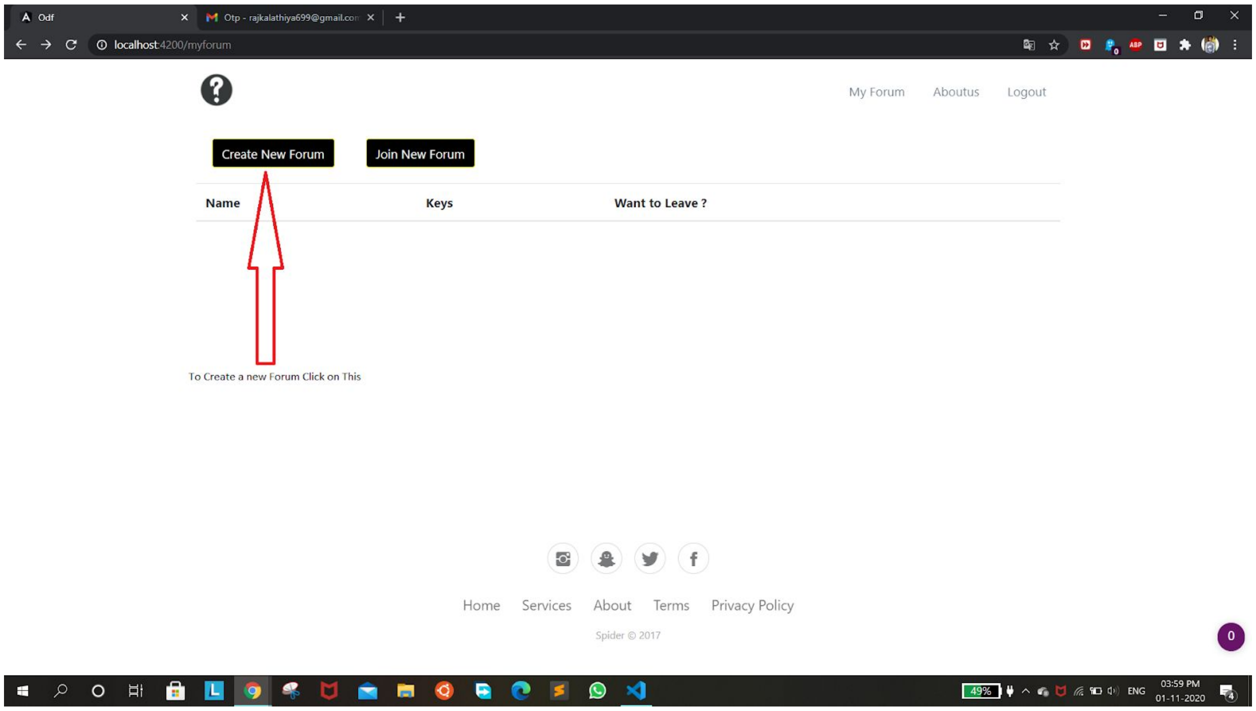
Password :-

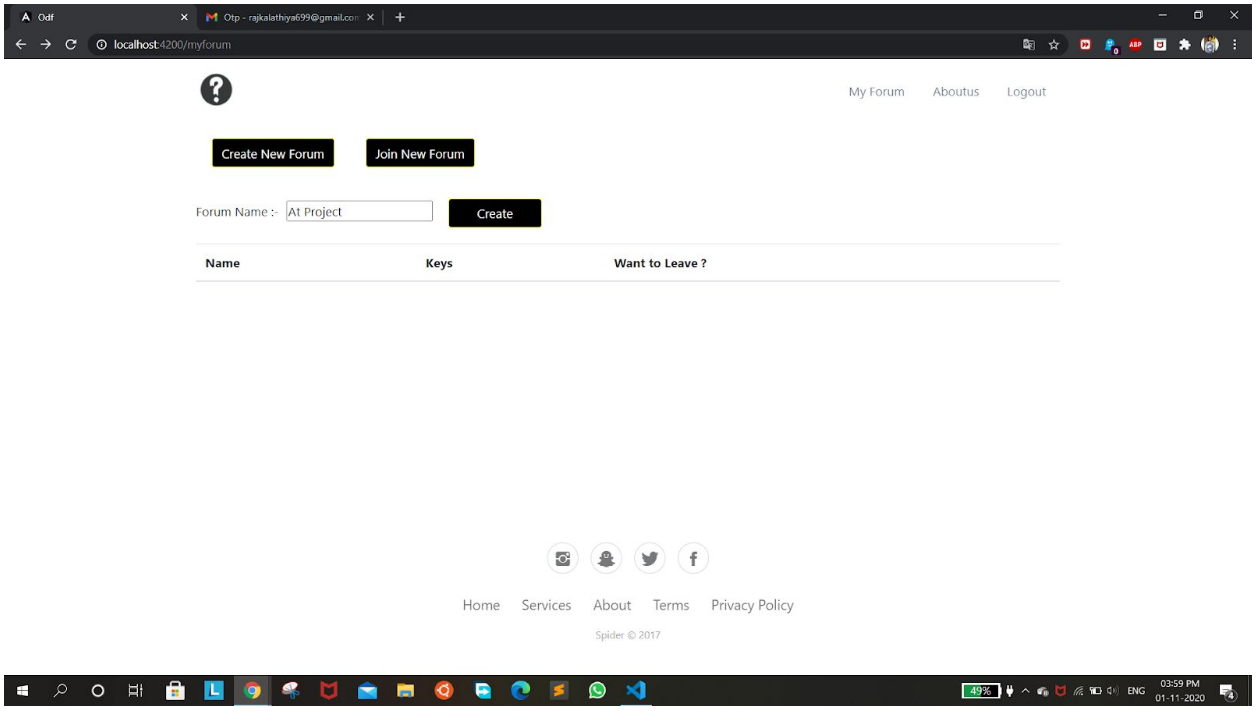
Enter Your password

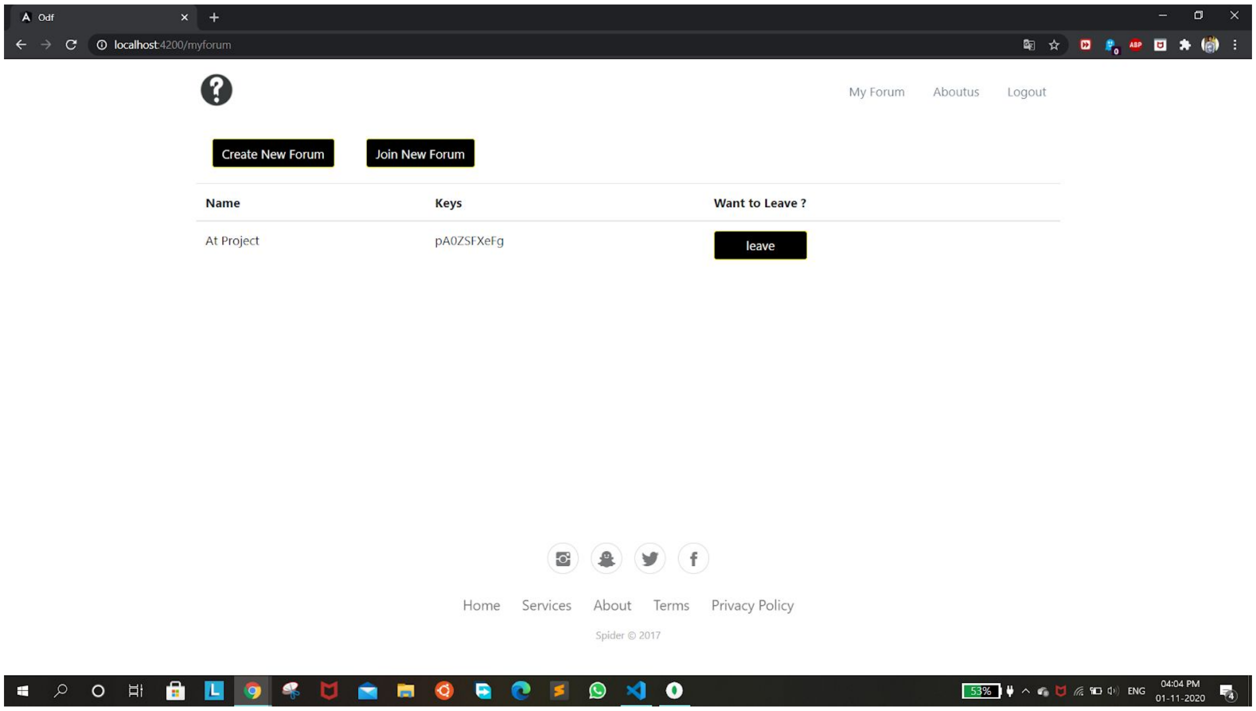
LogIn

Register new account

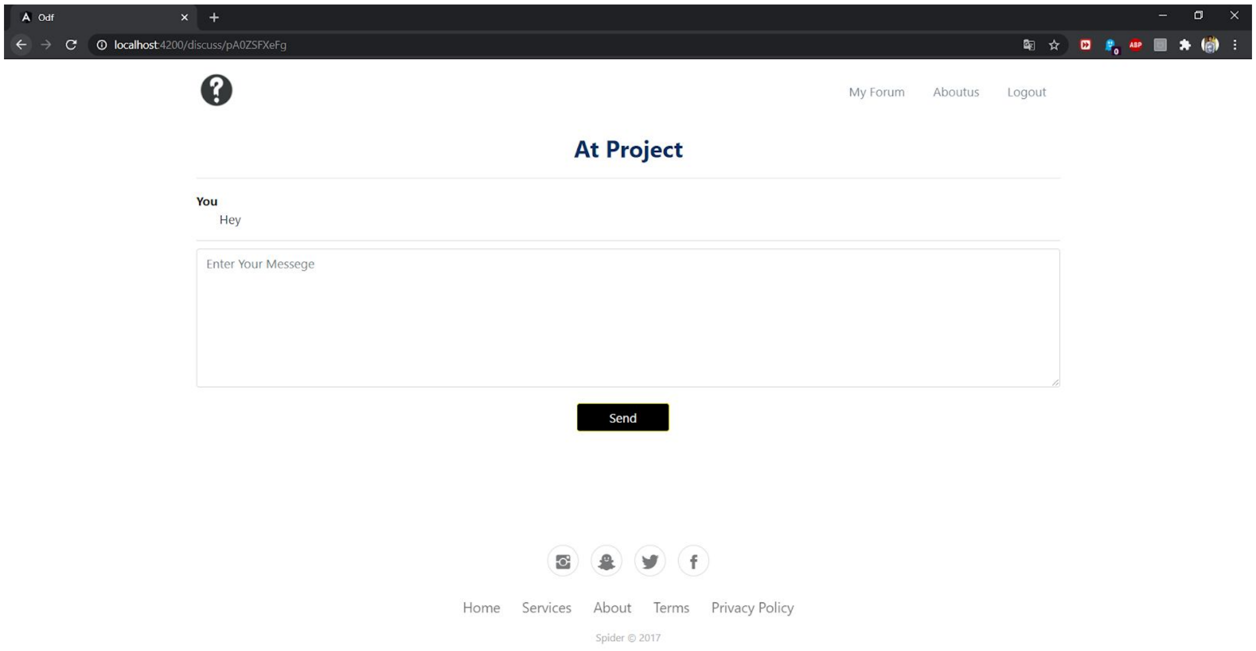
Forget Password


Odf
Otp - rajkalathiya699@gmail.com
localhost:4200/myforum
My Forum
Aboutus
Logout
Create New Forum
Join New Forum
Name
Keys
Want to Leave ?
To Create a new Forum Click on This
Home
Services
About
Terms
Privacy Policy
Spider © 2017


Odf
Otp - rajkalathiya699@gmail.com
localhost:4200/myforum
My Forum
Aboutus
Logout
Create New Forum
Join New Forum
Forum Name :- At Project
Create
Name
Keys
Want to Leave ?
Home
Services
About
Terms
Privacy Policy
Spider © 2017

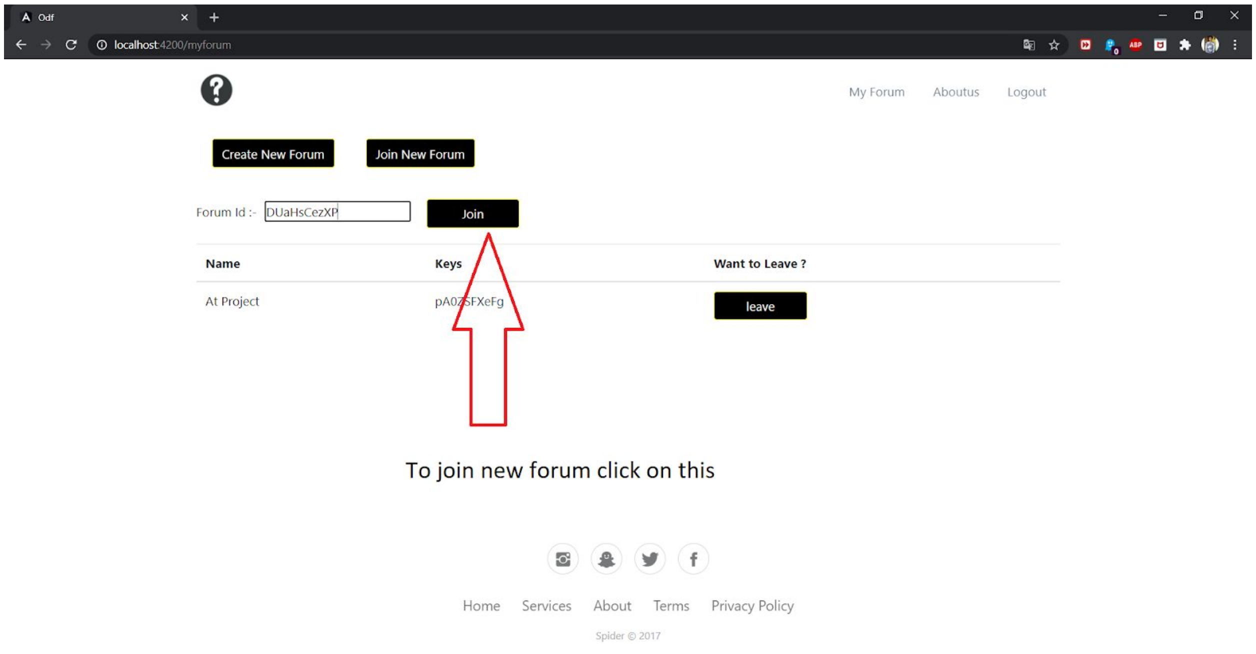


Name	Keys	Want to Leave ?
At Project	pA0ZSFXeFg	leave



At Project

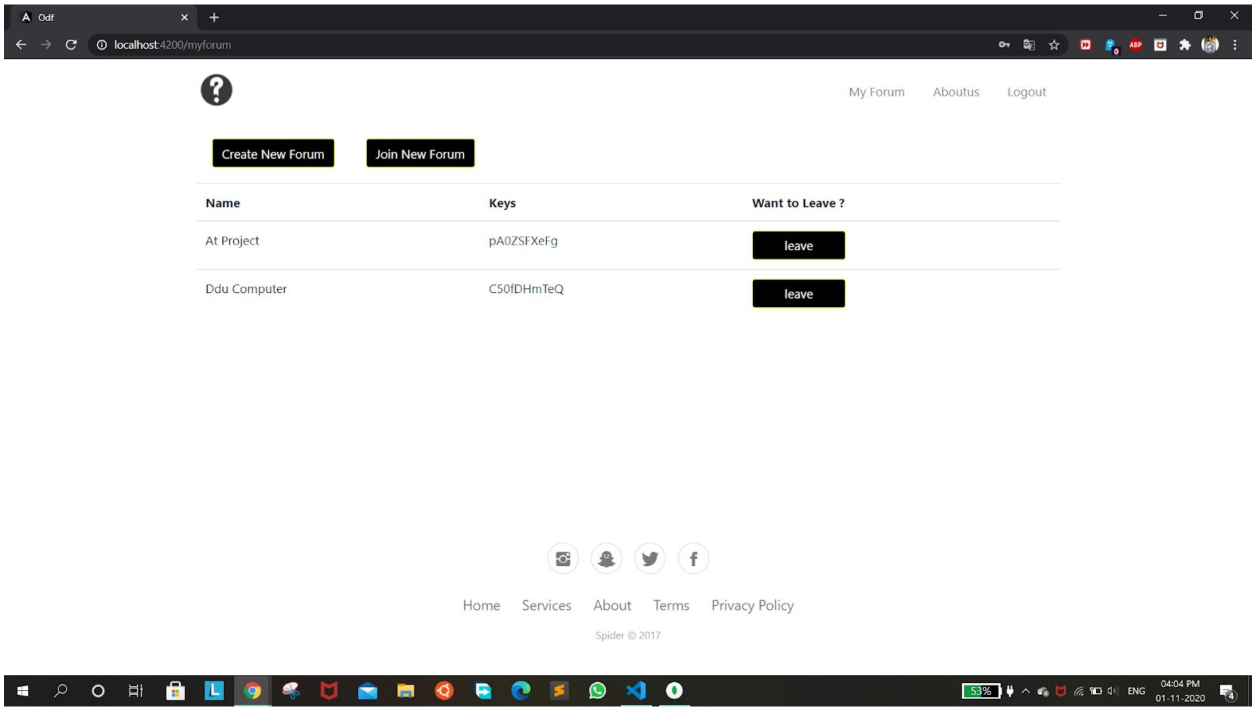
You
Hey



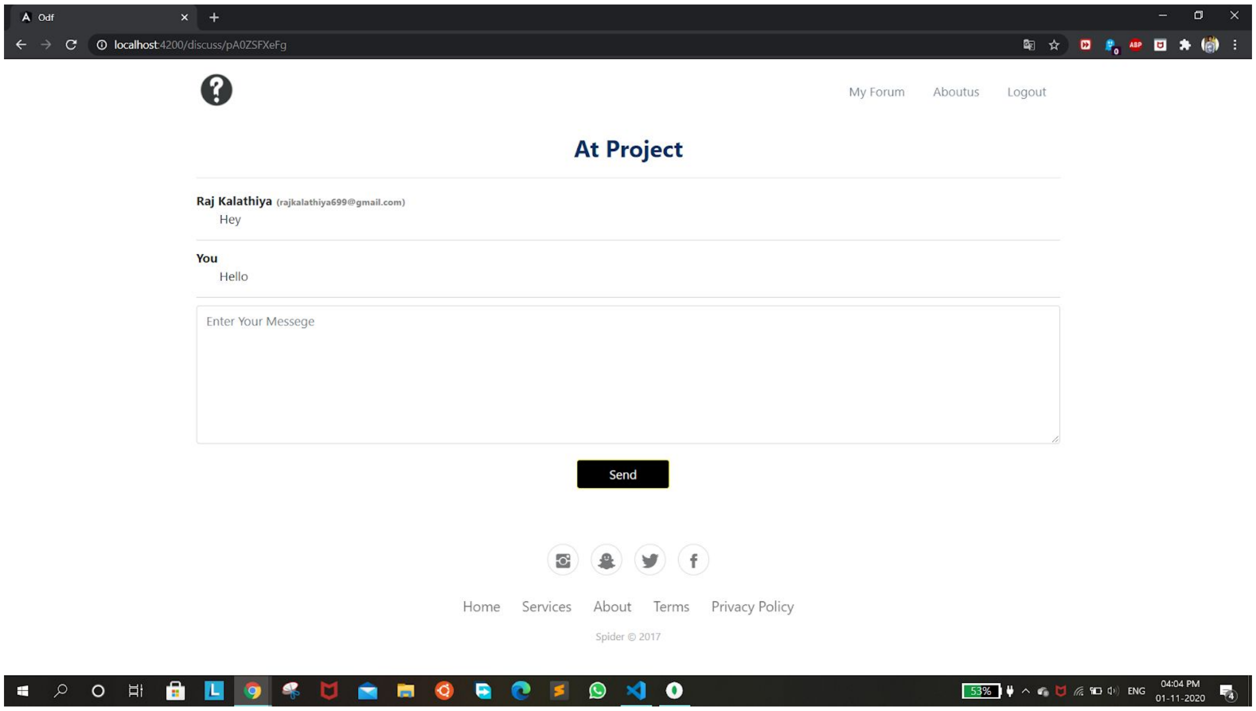
Forum Id :- DUaHsCezXP

Name	Keys	Want to Leave ?
At Project	pA0ZSFXeFg	leave

To join new forum click on this



Name	Keys	Want to Leave ?
At Project	pA0ZSFXeFg	leave
Ddu Computer	C50fDHmTeQ	leave



Odf

localhost:4200/discuss/pA0ZSFXeFg

My Forum Aboutus Logout

At Project

Raj Kalathiya (rajkalathiya699@gmail.com)
Hey

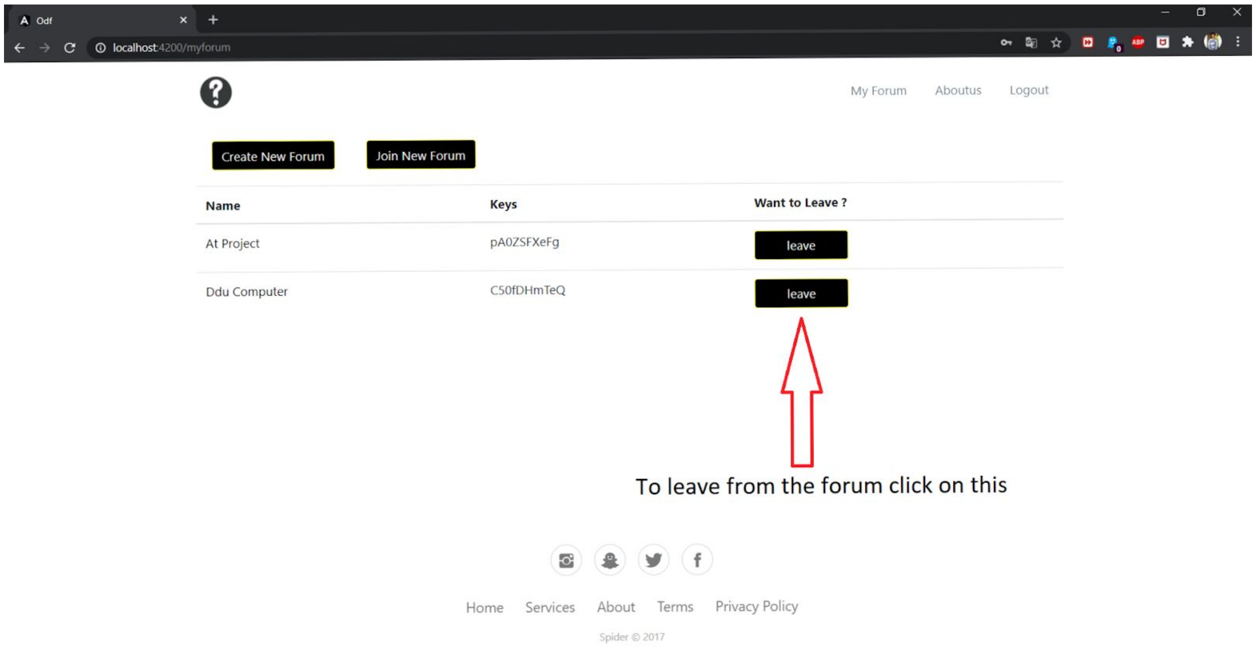
You
Hello

Enter Your Messege

Send

Home Services About Terms Privacy Policy

Spider © 2017



Odf

localhost:4200/myforum

My Forum Aboutus Logout

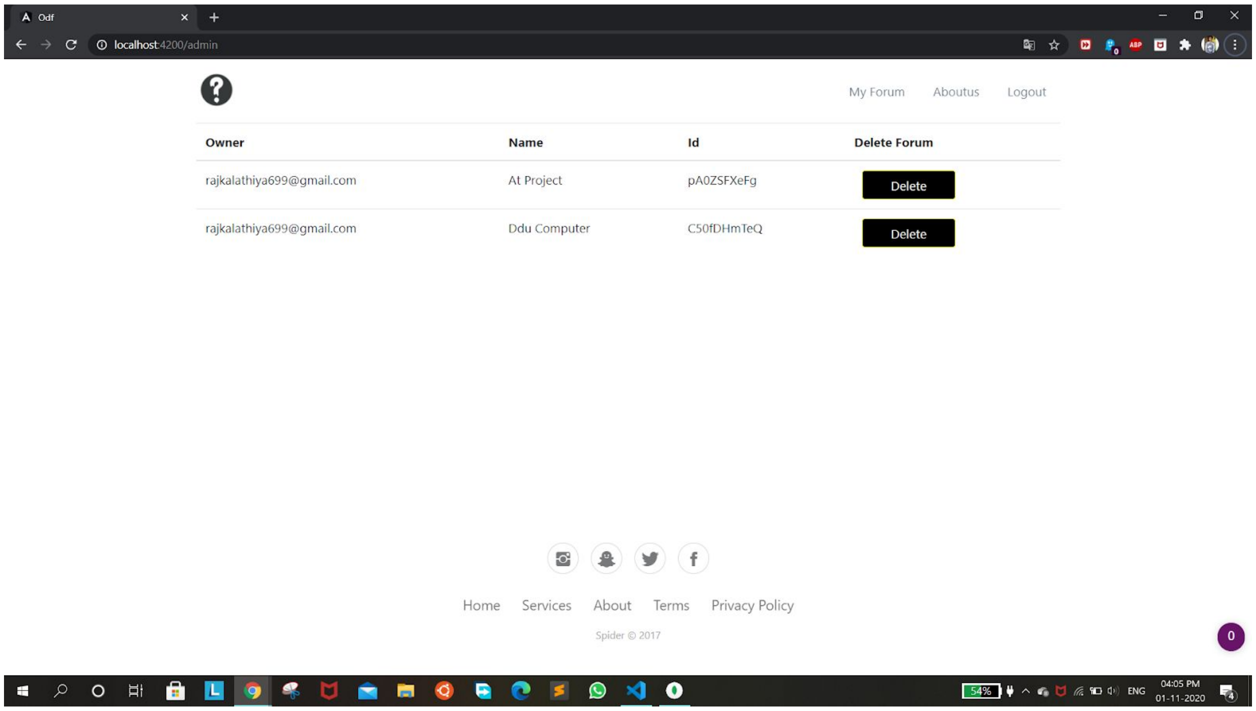
Create New Forum Join New Forum

Name	Keys	Want to Leave ?
At Project	pA0ZSFXeFg	leave
Ddu Computer	C50fDHmTeQ	leave

To leave from the forum click on this

Home Services About Terms Privacy Policy

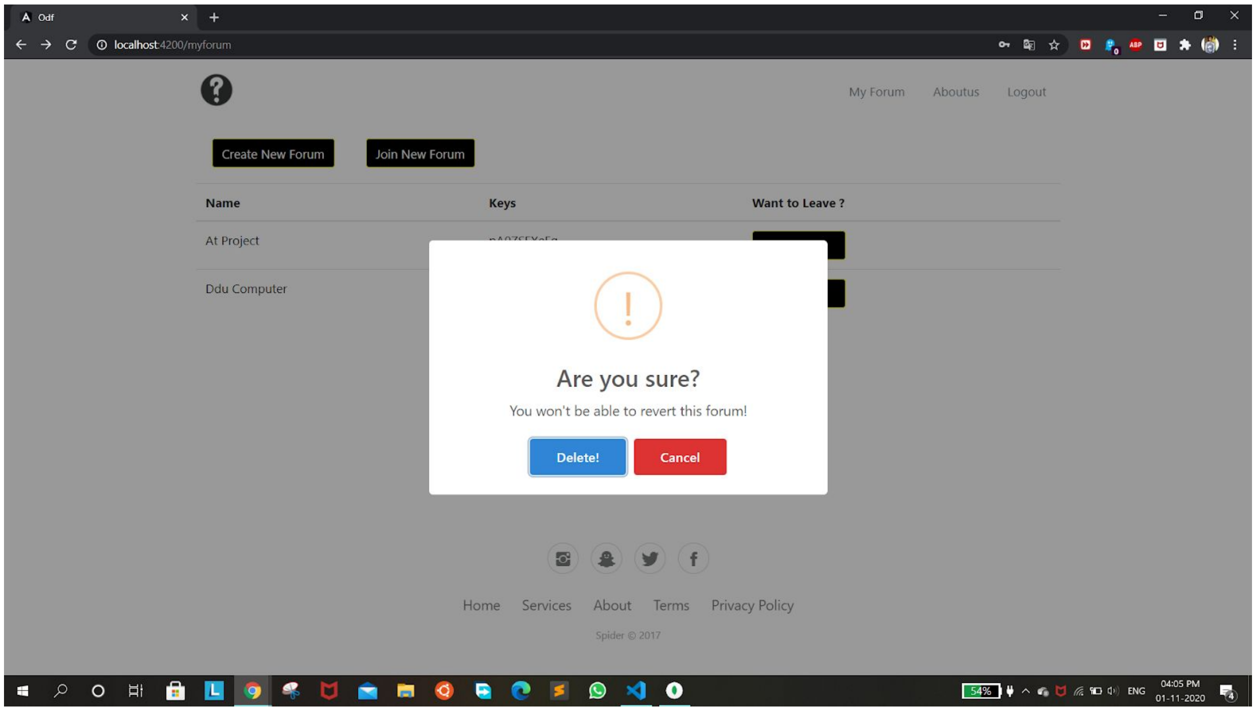
Spider © 2017



localhost:4200/admin

My Forum Aboutus Logout

Owner	Name	Id	Delete Forum
rajkalathiya699@gmail.com	At Project	pA0ZSFXeFg	Delete
rajkalathiya699@gmail.com	Ddu Computer	C50fDHmTeQ	Delete



localhost:4200/myforum

My Forum Aboutus Logout

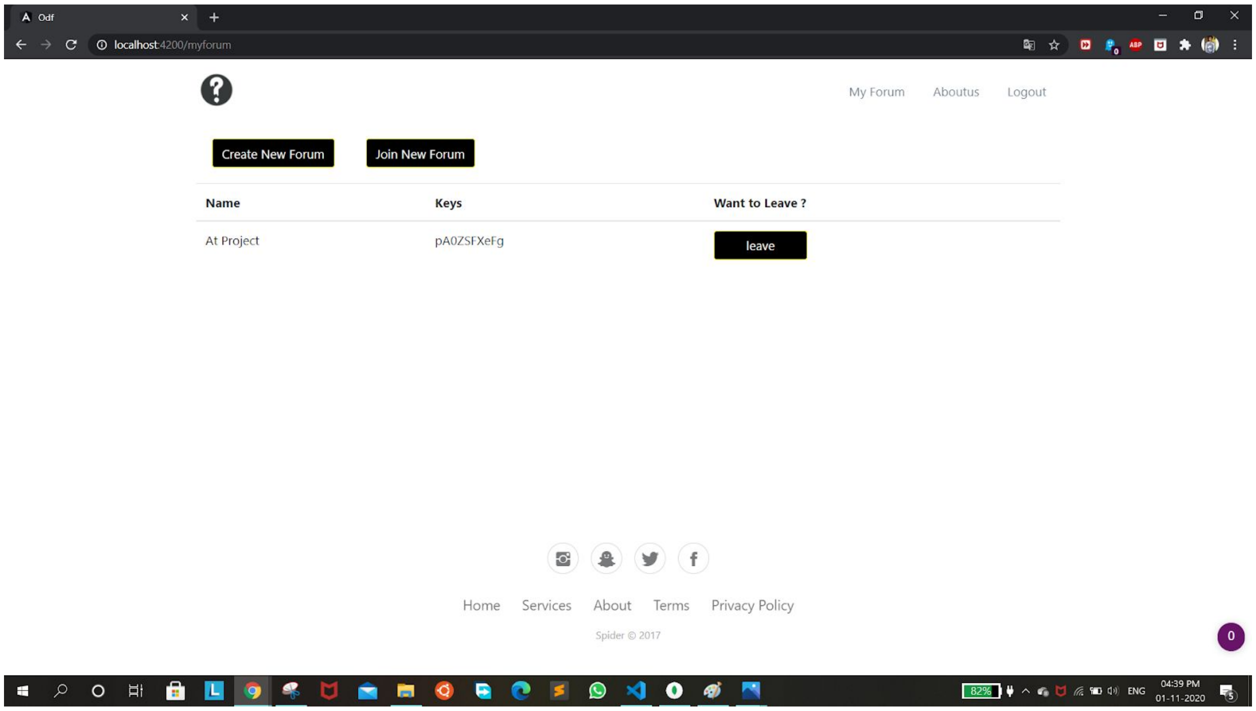
Create New Forum Join New Forum

Name	Keys	Want to Leave ?
At Project		
Ddu Computer		

Are you sure?

You won't be able to revert this forum!

Delete! Cancel



Name	Keys	Want to Leave ?
At Project	pA0ZSFXeFg	leave

- This is major functionality but there are several other functionality also.

Conclusion:

All the crud operations were done successfully on the user. If login details are not valid then the user can’t create or join the forums.if the user forgot password then he can change the password. Admin can delete the user and forum if found it suspicious.

- Create Forum
- Join Forum
- Leave from the forum
- Create a new account

Limitation and future extension:

Limitation:

Users can’t delete the message after they have sent it and they can’t send photos and files in the forum. If the user is the owner and he wants to remove another user from that form then the user can’t do that.

The passWord hashing mechanism is not used.

Future Extension:

Users which are in the same forum can see each other’s profile. They can send photos and files to forum users and can update their email id. Users can add the messages in their favourite messages. If a user has created one forum then he/she can remove another user from that forum. Improvement in Ui. If a user wants to join other’s forum then he must be added by the owner first.

Bibliography:

W3 School

HTML5: https://www.w3schools.com/html/default.asp

CSS: https://www.w3schools.com/css/default.adp

JavaScript: https://www.w3schools.com/js/default.asp

Jquery: https://www.w3schools.com/jquery/default.asp

XML: https://www.w3schools.com/xml/default.asp

XSLT: https://www.w3schools.com/xml/xsl_intro.asp

XSD: https://www.w3schools.com/xml/schema_intro.asp

JSON: https://www.w3schools.com/js/js_json_intro.asp

Bootstrap4:

https://getbootstrap.com/docs/4.3/getting-started/introduction/

Icons:

https://fontawesome.com/icons?d=gallery

Express.js:

https://expressjs.com/en/api.html

Node.js:

https://nodejs.org/en/

Nodemailer:

https://nodemailer.com/about/

MongoDB:

https://www.mongodb.com/

Sweetalert:

https://sweetalert2.github.io/

Other References:

https://stackoverflow.com/
https://www.youtube.com/