

# Webdev Project

Prepared by: Zhenyuan Xi, Yuan Cai, Yiran Zhang

April 18, 2019

---

# DESCRIPTION

## Overview

Our website is an online discussion site where people can hold conversations in the form of posted messages. Our users can 1. Check others' posts, 2. Comment on the posts they like, 3. Join our website community to post whatever they want to share.

## Roles

We have four roles since our team has three members.

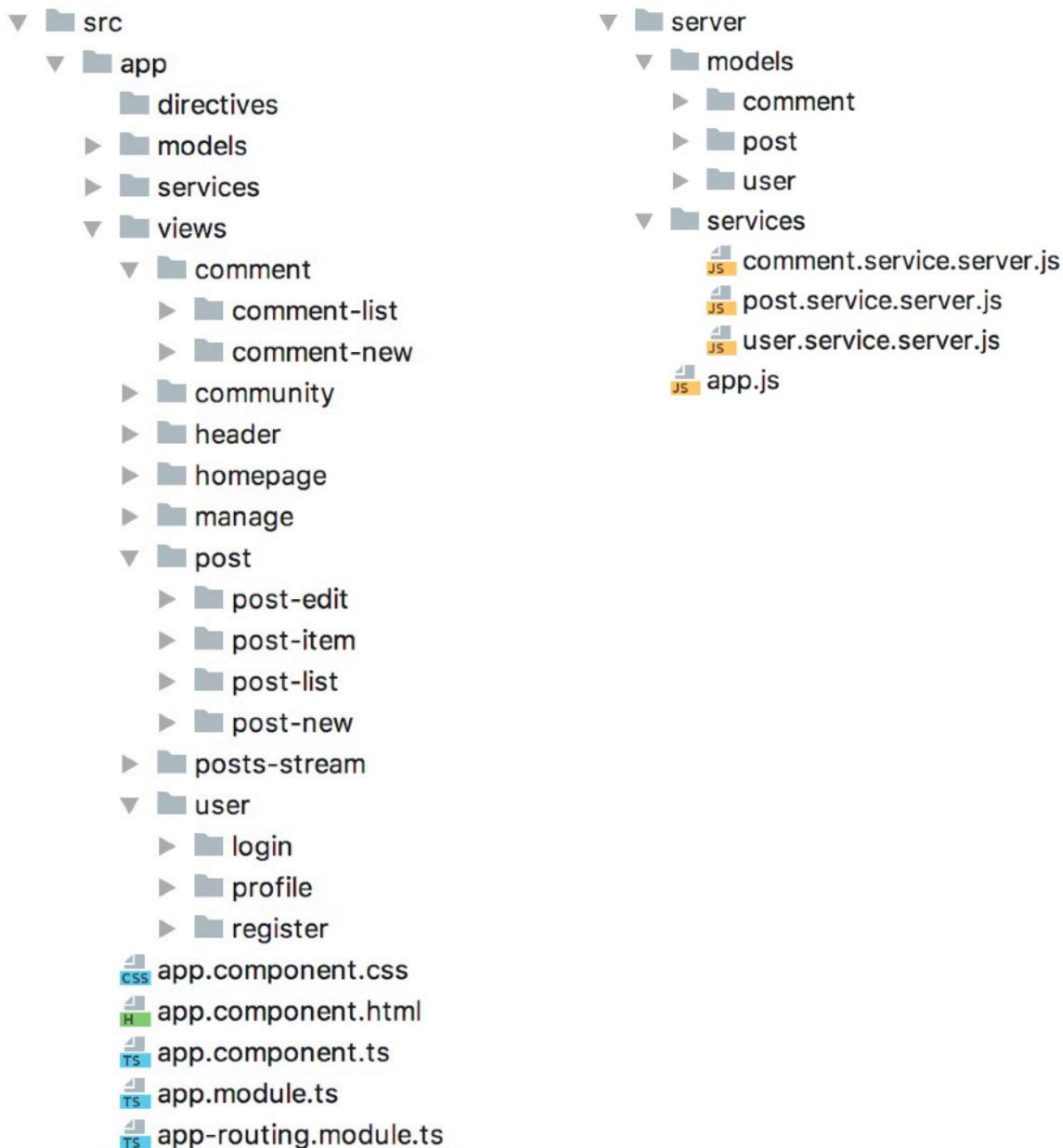
- Admin: admin can create new posts, check everyone's posts, comment on every post, delete non-admin users, posts or comments, have a specific page to manage all the data in the database. Admin cannot join community any more so they have no access to community page.
- User: user can check everyone's posts, comment on every post. They cannot create new posts and delete users, posts or comments. They have a community page for them to join our community to create new posts.
- Community member: member can create new posts, check everyone's posts, comment on every post. They cannot delete users, posts or comments.
- Guest: guest can only have access to our homepage and posts stream page to see the simplified content of each posts. They need to register to become a user.

## Tech Stack

- Use Angular7 for client side front-end development
  - Use Node.js and Express for server side back-end development
  - Use MongoDB as database
  - Use Passport LocalStrategy and Facebook Strategy for user authorization
  - Use Bcrypt for user password encryption and decryption
  - Use Google Map JavaScript API and agm (angular google map) to implement a google map service
  - Use Faker.js for a better user-friendly design where it can automatically generate images for posts
-

---

## Project Outline



## Routing

```
const routes: Routes = [
  { path: '', redirectTo: 'home', pathMatch: 'full' },
  { path: 'home', component: HomepageComponent },
  { path: 'login', component: LoginComponent },
  { path: 'register', component: RegisterComponent },
  { path: 'profile', component: ProfileComponent, canActivate: [AuthGuard] },
  { path: 'stream', component: PostsStreamComponent, canActivate: [AuthGuard] },
  { path: 'guest', component: PostsStreamComponent, canActivate: [GuestGuard] },
  { path: 'user/posts', component: PostListComponent, canActivate: [AuthGuard] },
  { path: 'stream/new', component: PostNewComponent, canActivate: [AuthGuard] },
  { path: 'stream/:pid', component: PostItemComponent, canActivate: [AuthGuard] },
  { path: 'stream/:pid/comment', component: CommentListComponent, canActivate: [AuthGuard] },
  { path: 'stream/:pid/comment/new', component: CommentNewComponent, canActivate: [AuthGuard] },
  { path: 'community', component: CommunityComponent, canActivate: [AuthGuard] },
  { path: 'manage', component: ManageComponent, canActivate: [AuthGuard, AdminGuard] },

  { path: '**', redirectTo: 'home' }
];
```

## Object Design

- Use a 'type' attribute for 'user' objects to differentiate them where type could be admin, user or membership.

```
var userSchema = mongoose.Schema({
  facebook: {
    id: String,
    token: String
  },
  username: String,
  password: String,
  firstName: String,
  lastName: String,
  email: String,
  address: String,
  state: String,
  country: String,
  type: String,
  posts: [postSchema],
  dateCreated: {type: Date, default: Date.now()}
}, {collection: 'users'});
```

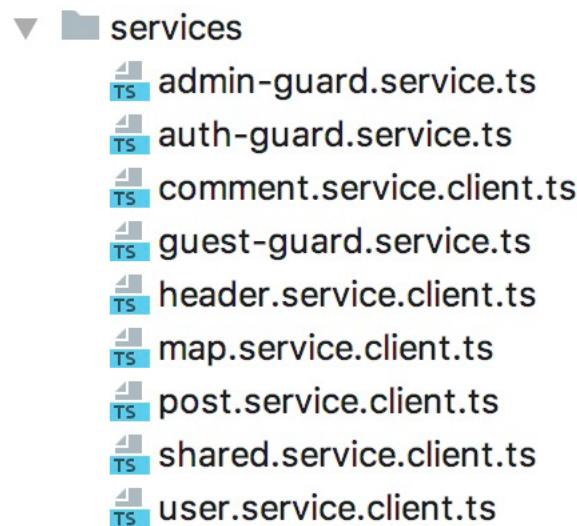
- Create 'post' object and reference it to user.\_id.

```
var postSchema = mongoose.Schema({
  userId: {type: mongoose.Schema.Types.ObjectId, ref: "User"},
  title: String,
  content: String,
  imagePath: String,
  username: String,
  comments: [commentSchema],
  dateCreated: {type: Date, default: Date.now()}
}, {collection: 'posts'});
```

- Create 'comment' object and reference it post.\_id.

```
var commentSchema = mongoose.Schema({
  postId: {type: mongoose.Schema.Types.ObjectId, ref: "Post"},
  username: String,
  content: String,
  dateCreated: {type: Date, default: Date.now()}
}, {collection: 'comments'});
```

- Protect specific data according to the 'user' object's type and use some guard services to protect components.



---

## Object Relationship

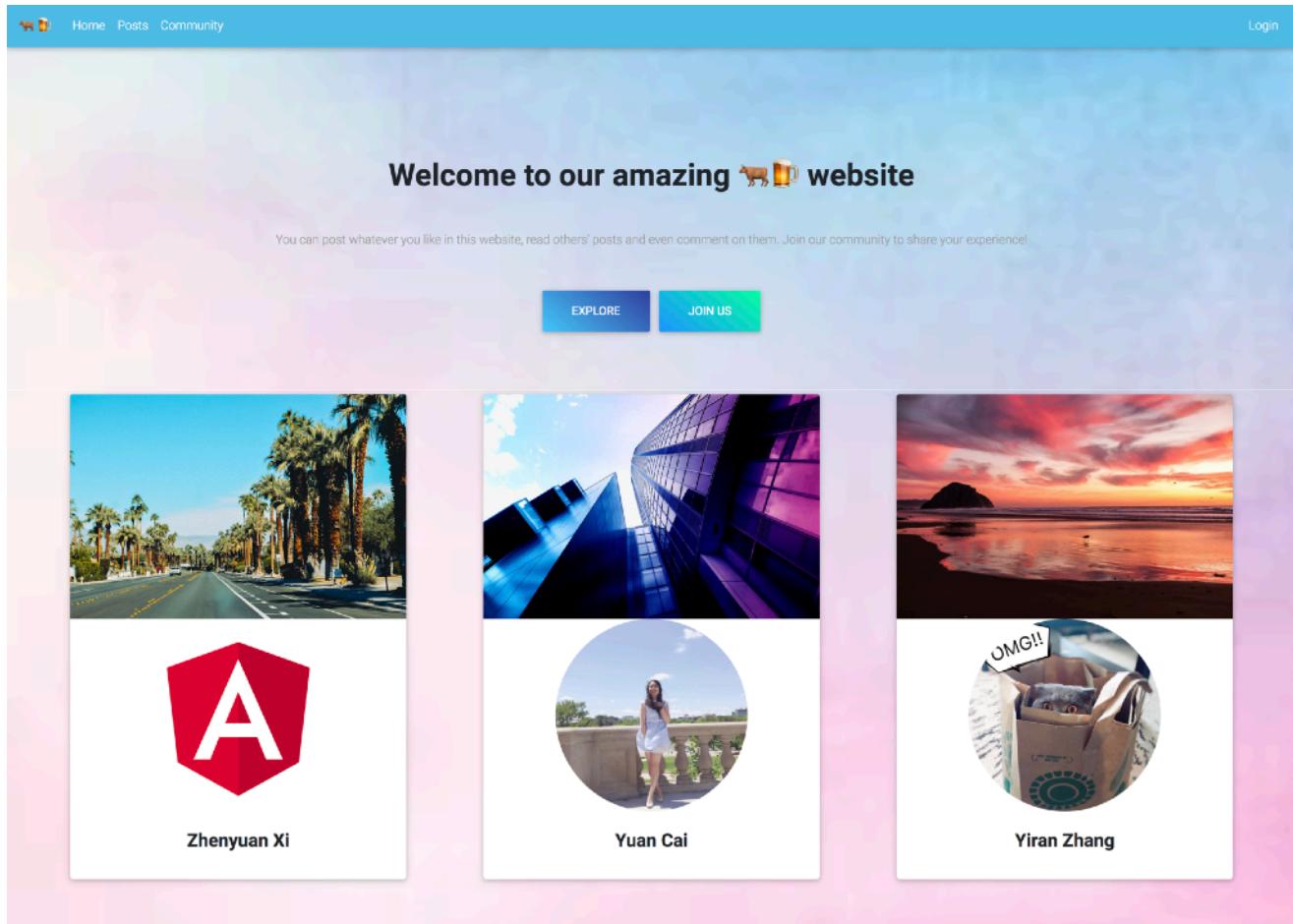
- Admin
  - ✓ Header, Home, Manage, all User, Posts-stream, all Post, all Comment components
  - ✗ Community component
- User
  - ✓ Header, Home, Community, all User, Posts-stream, Post except post-new, all Comment components
  - ✗ post-new, Manage components
- Community Member
  - ✓ Header, Home, Community, all User, Posts-stream, all Post, all Comment components
  - ✗ Manage component
- Guest
  - ✓ Header, Home, all User, Posts-stream components
  - ✗ Community, Manage, all Post, all Comment components

## Potential Improvement

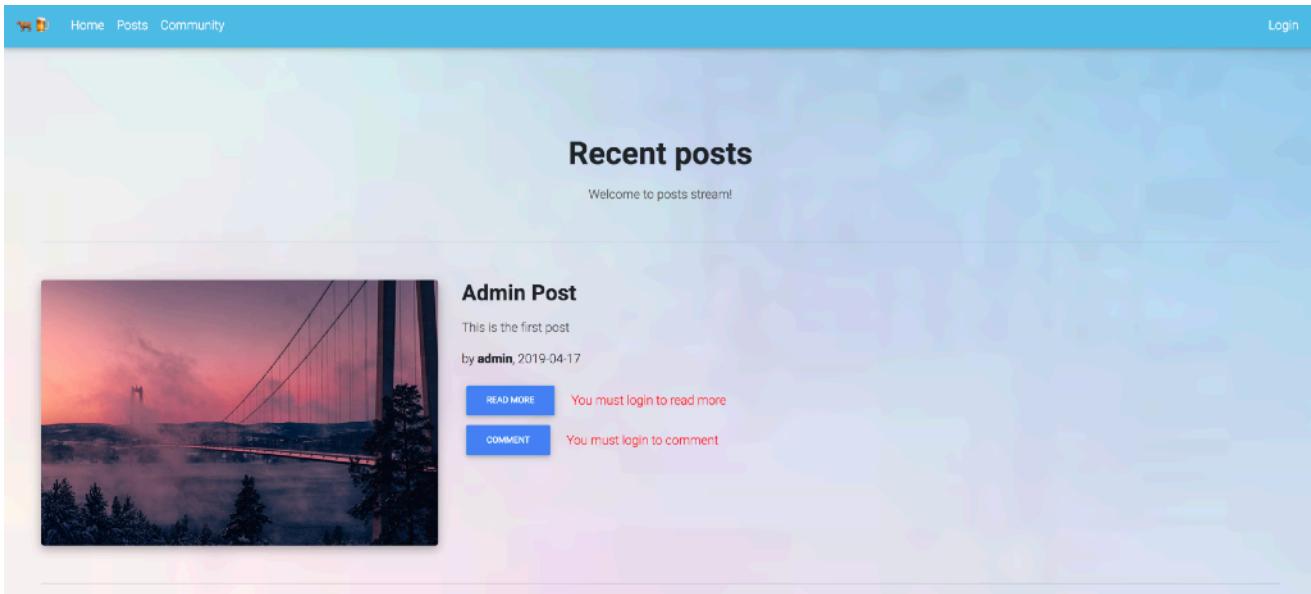
- Add a page for membership user to manage their membership and add the payment if possible.
  - Add pages for more interaction for users, enable them to add friends and talk with each other privately.
  - Add more detailed and editable pages for each post to make it better looking and informative.
  - Add more security authorities for admin user and a different way to register as an admin user.
  - Add more topics to post and let people the things they are interested in specifically.
-

# TEST

## Welcome to our website as a guest



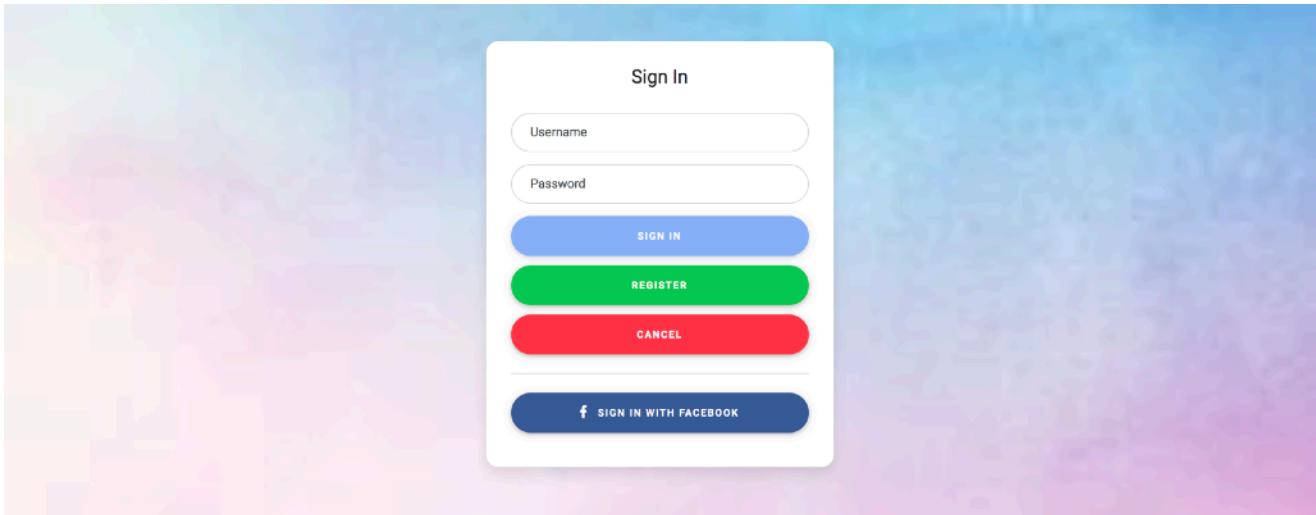
- Click EXPLORE to explore our website as a guest.
- Click JOIN US to register and login, also you could do it by clicking the top-right Login button.
- Click other buttons on the header will redirect you to login page.



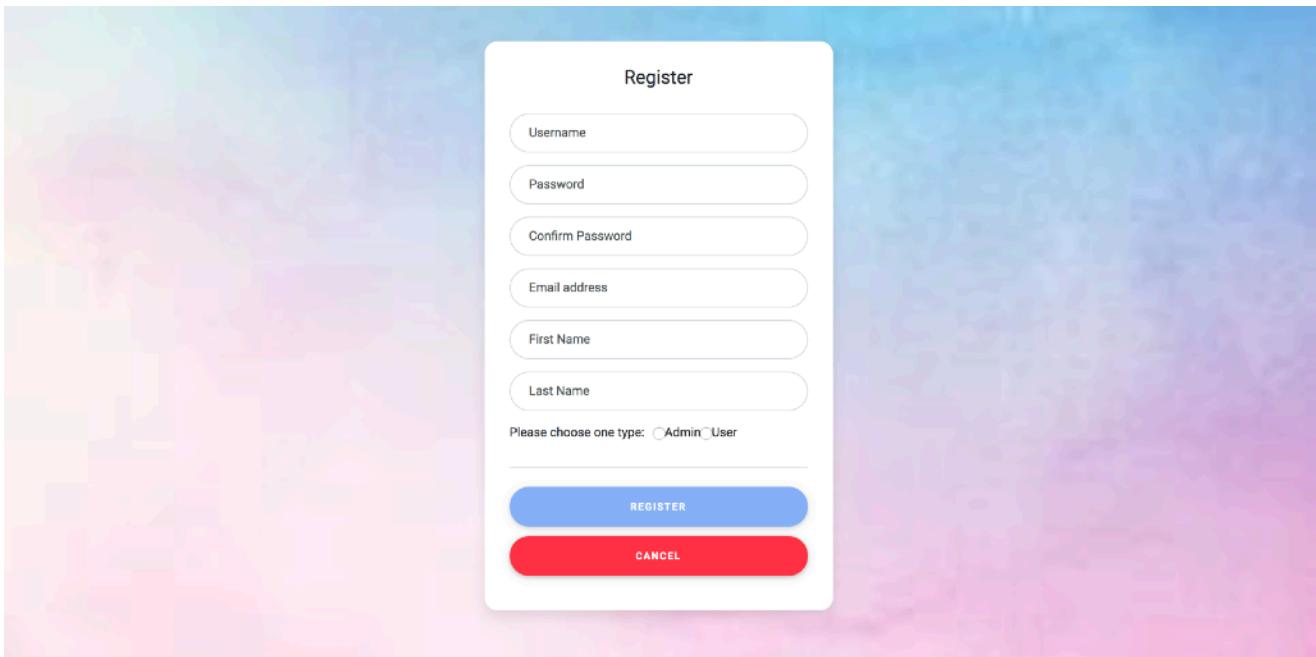
- It's our posts stream page for guest user, guest needs to login for further access.
- Click any button in this page will redirect to login page.

---

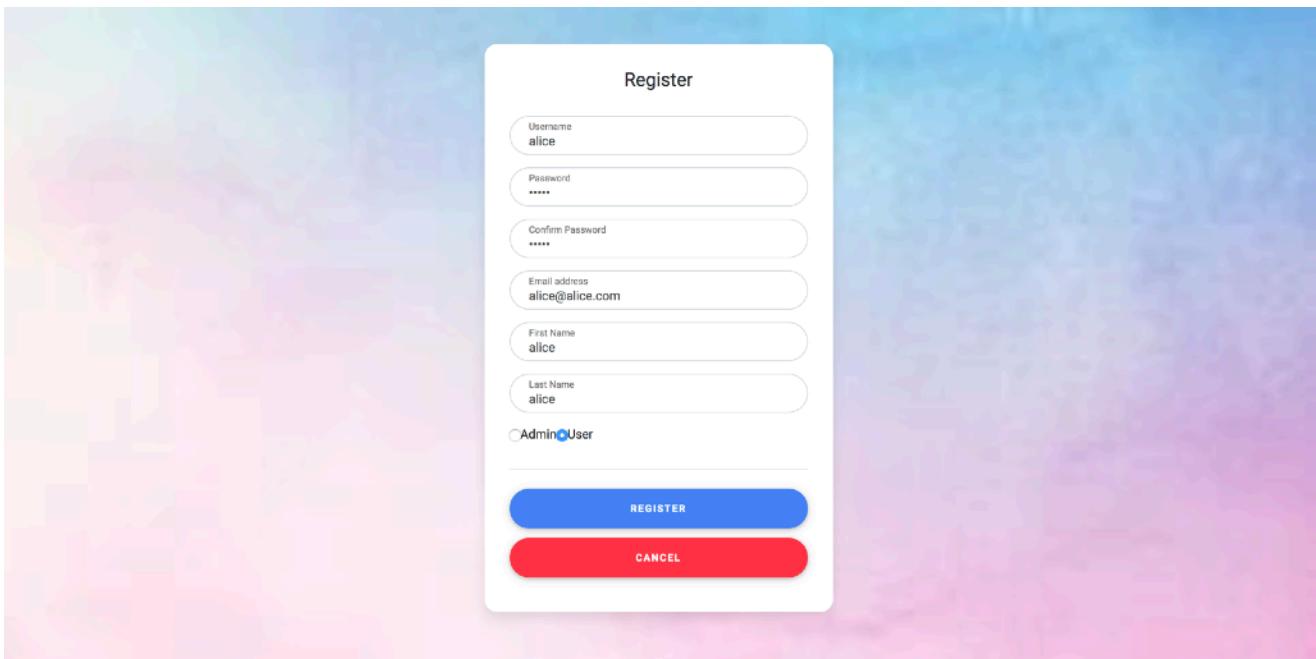
## Register and login



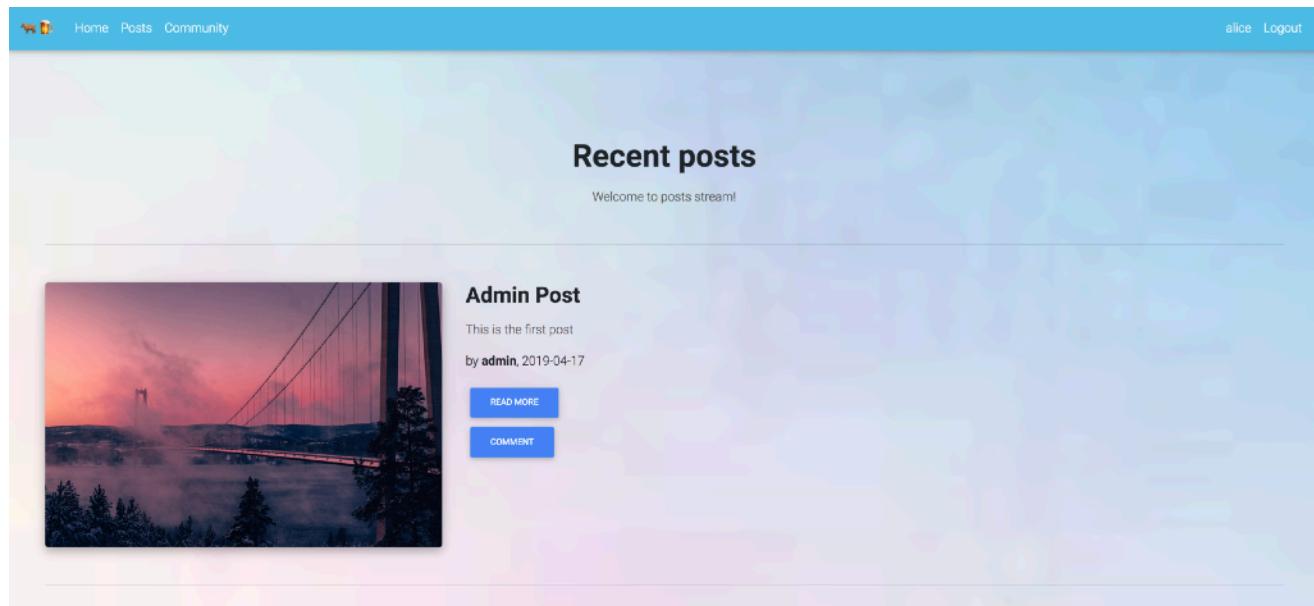
- It's our login page, you can sign in if you enter correct username and password.
  - You can register as a new user, or cancel to go back to the home page.
  - You can sign in with your Facebook account.
-



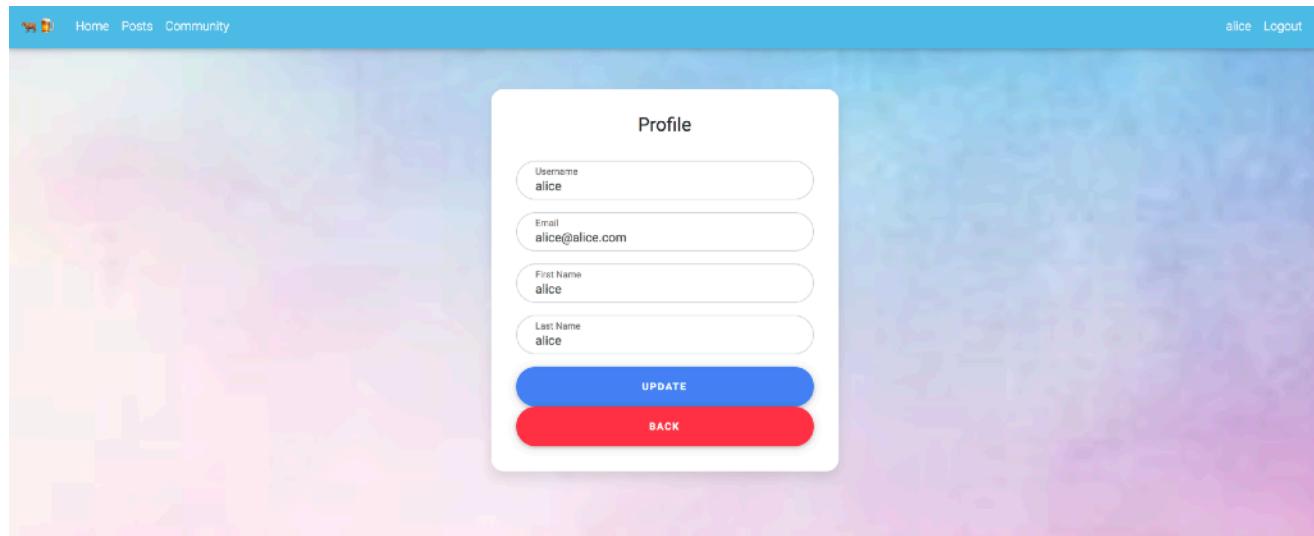
- You can register with required username, password, confirm password and email address.
- Choose to be a normal user or an admin user, we may add further functionality for becoming an admin.
- Now register as 'alice' and do further tests (password is 'alice').



**Explore our website posts and leave your comments**



- Now you login as 'alice', you can click your username in the top-right to check your profile and update it.
- You can read more details about the post and leave your comments.
- You can click Community in the header to join our community to become a membership.



Home Posts Community alice Logout



## Admin Post

Content  
This is the first post.

Author  
admin  
2019-04-17

BACK

Home Posts Community alice Logout

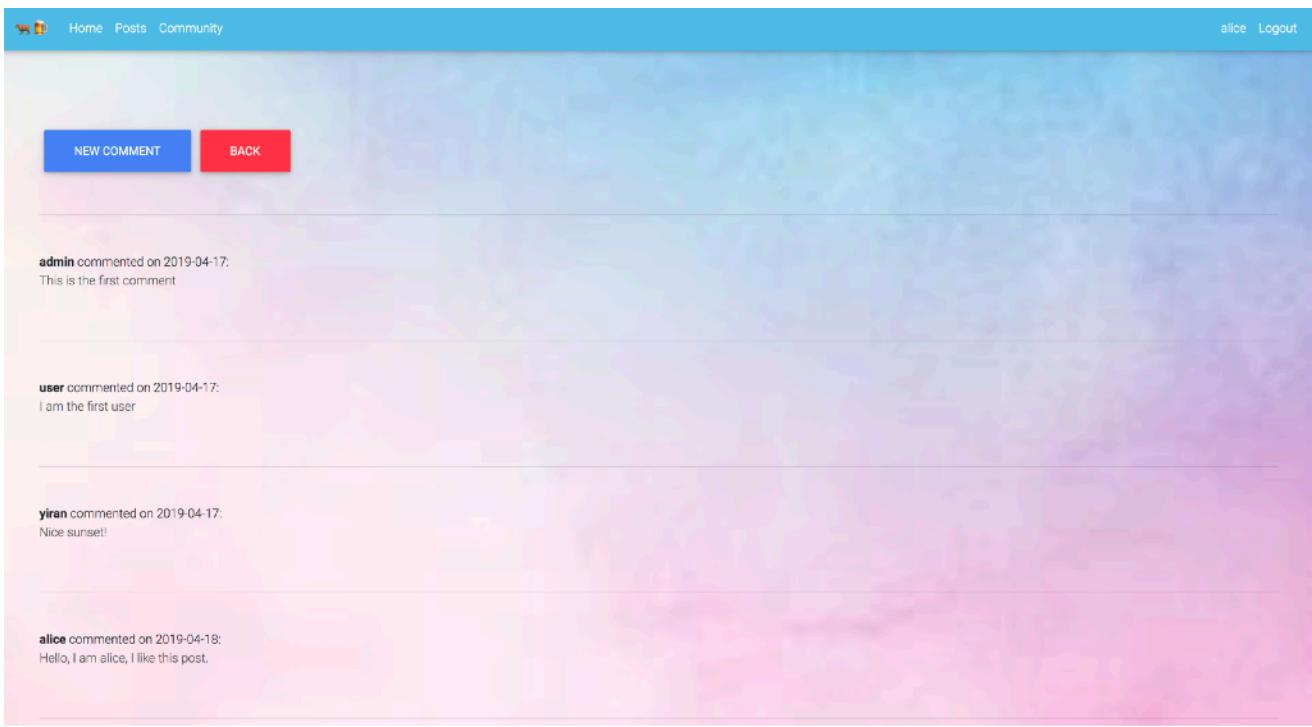
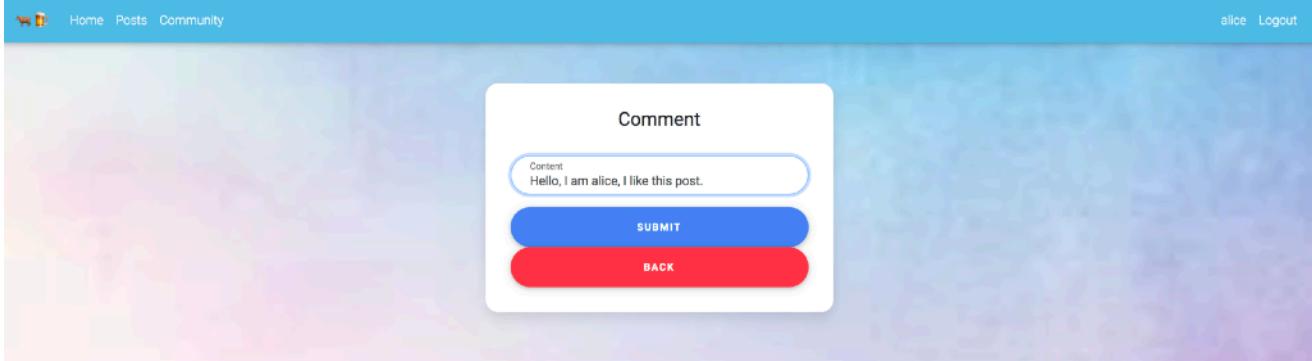
NEW COMMENT BACK

admin commented on 2019-04-17:  
This is the first comment

user commented on 2019-04-17:  
I am the first user

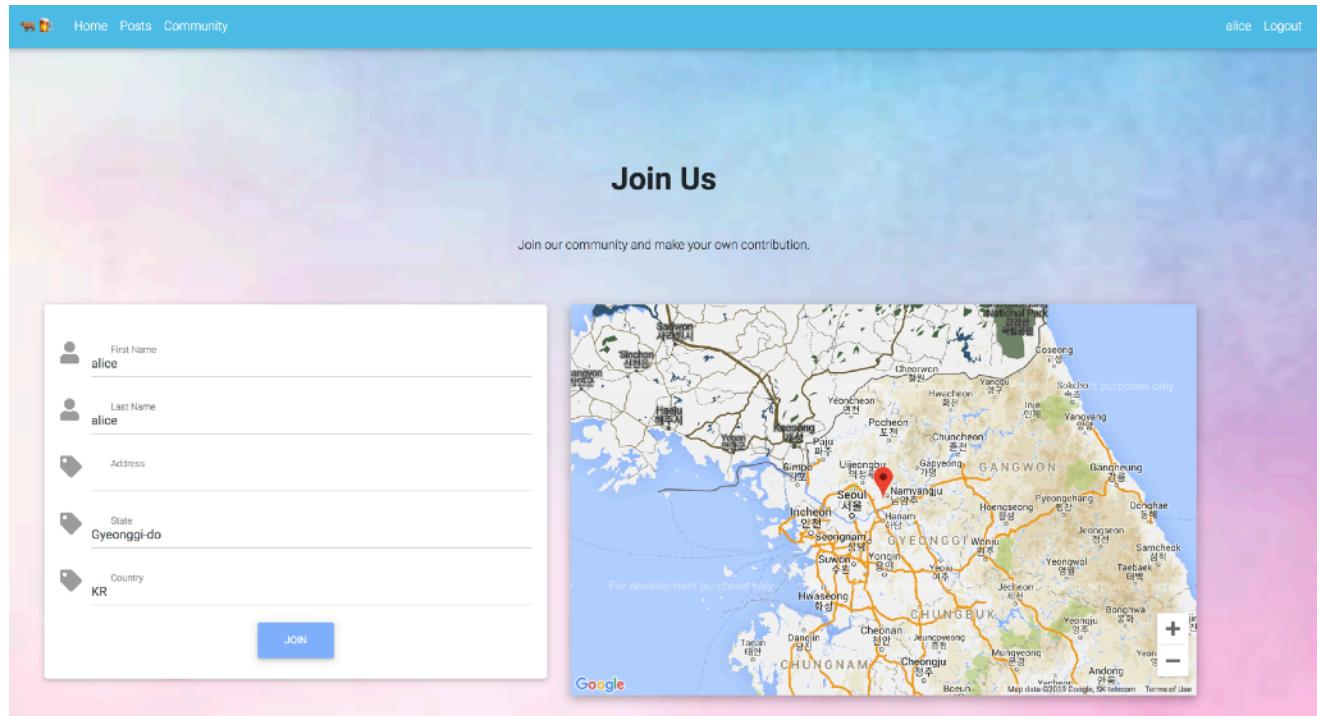
yiran commented on 2019-04-17:  
Nice sunset!

- You can see all other users' comments for this post.
- Leave your comments for this post now.

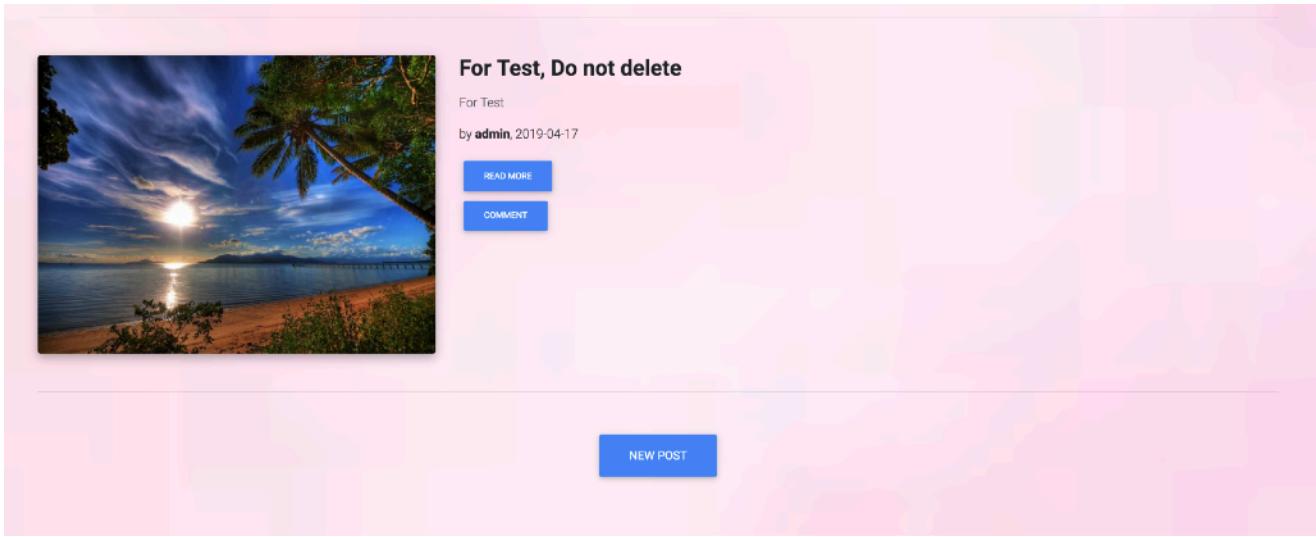


- Good Job! We got your first comment for this post.
- Now it's time to join our community to share your thoughts.

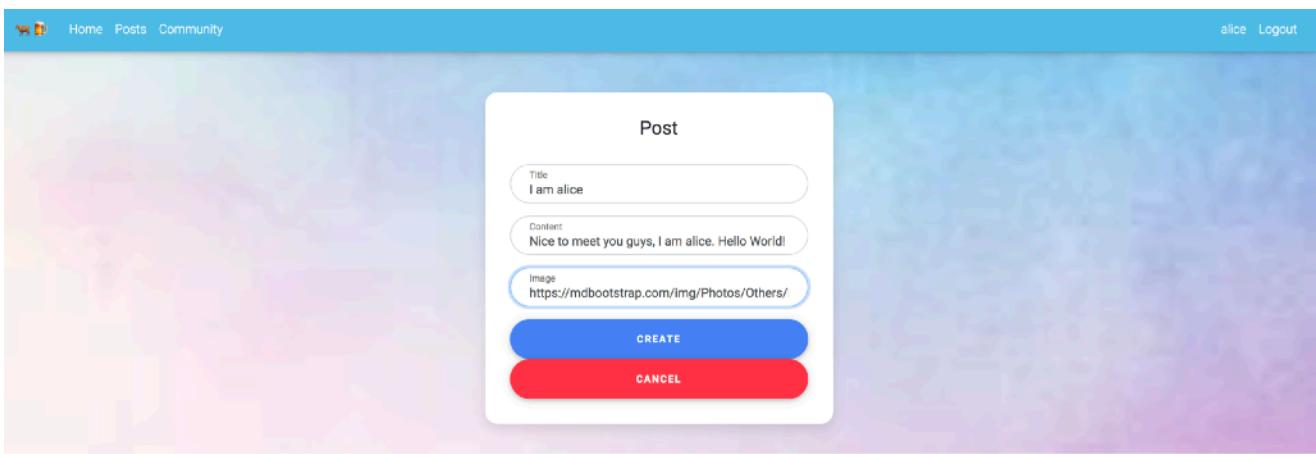
Join our community to post something



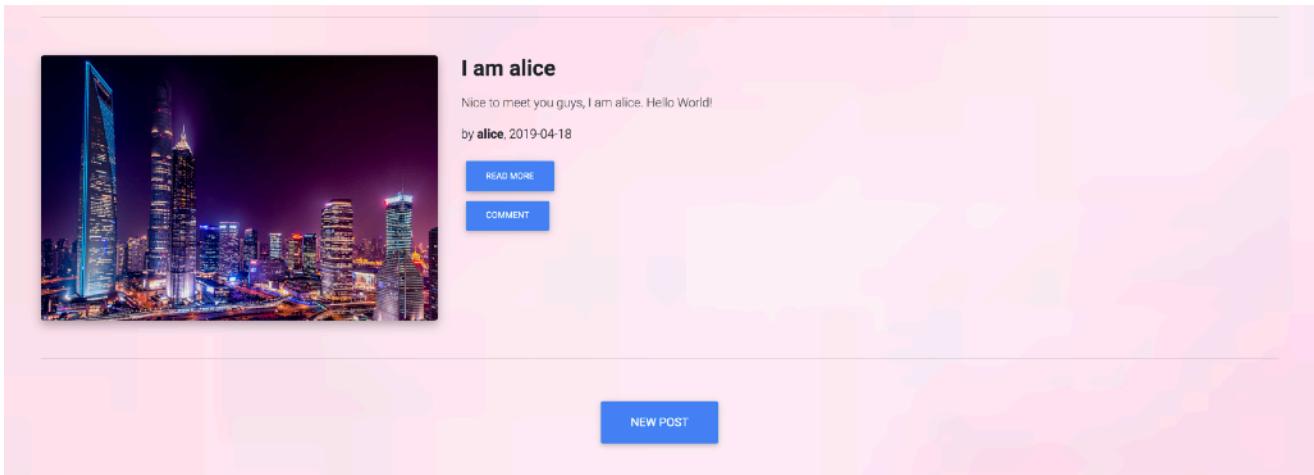
- It's our community page when you click the Community button.
- Use a IP fetch api and google map api, we can know your current address (why we are in Korea??? lmao).
- Click JOIN then you will become a membership and have access to post something.



- Now on the bottom of the posts stream page, you have a NEW POST button as a membership.
- Click and make your first post.



- Create your first post with the image URL (if not, we will generate a random one for you).
- Title and content are required



- Wow, your post are on the posts stream!
- The last thing is to become an admin and manage the whole website.

## Become an admin and help manage our website

Recent posts

Welcome to posts stream!

**Admin Post**

This is the first post

by admin. 2019-04-17

READ MORE

COMMENT

DELETE

- Use our example admin account to login (username: admin, password: admin).
- You will find a Manage button rather than a Community button
- You will have an extra option to delete others' post, as well as comments but not shown in this page.

BACK

## USERS

Username	First Name	Last Name	Email	State	Country	Address	Type	Option
user	user	user	user@user.com				User	<button>DELETE</button>
member	member	member	member@member.com	Washington	US	member	membership	<button>DELETE</button>
yiran	Yiran	Zhang	zhangyiran0401@gmail.com	Washington	US	401 Terry Ave N	membership	<button>DELETE</button>
admin	admin	admin	admin@admin.com				Admin	
maptest	maptest	maptest	maptest@maptest.com				User	<button>DELETE</button>
asd	asd	asd	asd@asd.com	Washington	US	asd	membership	<button>DELETE</button>
alice	alice	alice	alice@alice.com	Gyeonggi-do	KR	alice	membership	<button>DELETE</button>

## POSTS

Title	Content	Username	Image	Option
Admin Post	This is the first post	admin	<a href="https://mdbootstrap.com/img/Photos/Others/img%20(34).jpg">https://mdbootstrap.com/img/Photos/Others/img%20(34).jpg</a>	<button>DELETE</button>
Member Post	I am the first member to post something	member	<a href="https://mdbootstrap.com/img/Photos/Others/laptop-sm.jpg">https://mdbootstrap.com/img/Photos/Others/laptop-sm.jpg</a>	<button>DELETE</button>
Notre Dame	Notre Dame Cathedral after Fire	yiran	<a href="https://odn.theatlantic.com/assets/media/img/photo/2019/04/after-fire-photos-inside-notre-dame/n01_1137551534/main_1500.jpg?1555438166">https://odn.theatlantic.com/assets/media/img/photo/2019/04/after-fire-photos-inside-notre-dame/n01_1137551534/main_1500.jpg?1555438166</a>	<button>DELETE</button>
For Test, Do not delete	For Test	admin	<a href="http://corempixel.com/640/480/nature">http://corempixel.com/640/480/nature</a>	<button>DELETE</button>
I am alice	Nice to meet you guys, I am alice, Hello World!	alice	<a href="https://mdbootstrap.com/img/Photos/Others/images/83.jpg">https://mdbootstrap.com/img/Photos/Others/images/83.jpg</a>	<button>DELETE</button>

## COMMENTS

Content	Username	Option
This is the first comment	admin	<button>DELETE</button>
I am the first user	user	<button>DELETE</button>
It's the first comment from a member	member	<button>DELETE</button>
Nice sunset!	yiran	<button>DELETE</button>
your first post is awesome	admin	<button>DELETE</button>
For Test, Do not delete	admin	<button>DELETE</button>
Hello, I am alice, I like this post.	alice	<button>DELETE</button>

- 
- Look how you can do as our admin user LOL.
  - You can check everything and delete everything except yourself and other admin users.

### **That's all and logout**

- Click Logout button on the top-right at anytime and it will redirect to homepage.

# THANK YOU