THE HONG KONG POLYTECHNIC UNIVERSITY
香港理工大學

COMP5321 Enterprise Web and Internet Computing for Managers
Group Project: Personal Blog System

Name	Student ID
Chen Yongxin	18073824G
Zhao Haiqi	18087348G

Table of Content

Description

Our project is a Personal Blog System. Blog's admin can log in the system and can add/delete/edit the articles. Other users can browse the articles.

The main purpose of our project is to design and implement a personal blog system. Although there are many useful blog systems on the internet, they have less custom design and settings and not all of them support markdown editing. What's more, we implement our blog system using AngularFire framework that the system are more comprehensive. As a result, we can manage our own data in the cloud database. Every user keeps their accounts only. It protect the privacy compare to the existing blog system on the internet. In addition, we can design our blog style and layout whatever we like.

Above the main blog body, there is a character called Waddle Dee. We design an animation that Waddle Dee sways in the wind and comes down in a parachute, leading you to our blog when he lands in the ground. In the blog body, there are different categories, each category contains different articles. Users can view the full article, click the like button and make comments.

In the "add post" page, admin can add his/her own articles. Our blog system also supports markdown. Administrator can use markdown editor online to write the post. In addition, the post can be displayed in markdown format.

In each post, admin can edit or delete the post and update it to the cloud database when pressing the "edit" or "delete" button. Other users can like or comment the post.

Compared to other blog system on the internet, there is no advertisement and other social media contents in our blog system, which provides more neat and clear posts sharing and reading experience for user.

Project Details

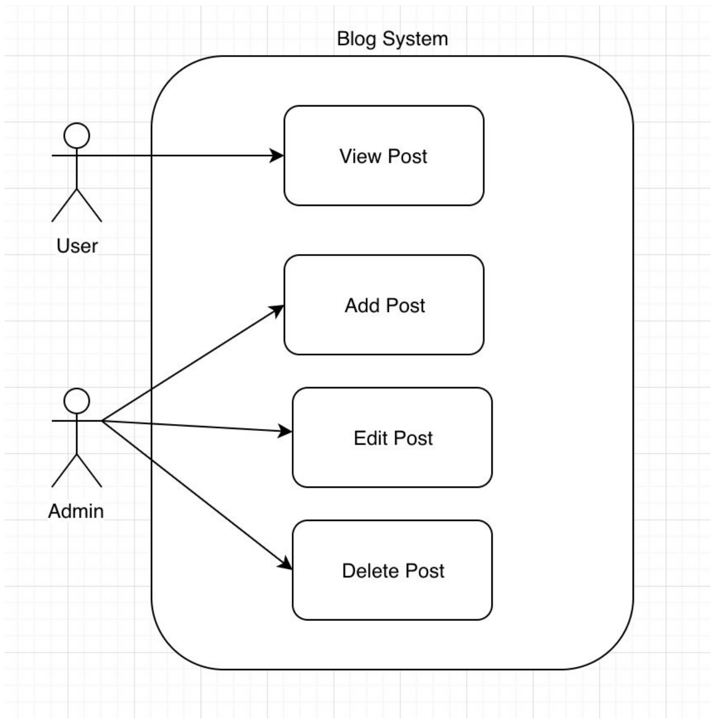
System Structure

There are some diagrams for the system structure.

- **Use case diagram**

Here is the use case diagram. It specifies that user can view the post in the our blog system, and the Admin can manage the post in the blog system.

Figure 1. Use case diagram



- **Flowchart**

Here is a system flowchart below. In this diagram, user opens our website, then scrolling down from the sky to the ground, then they can see our blog. At this point, if users want to look for post by category, they can click the button on the navigator bar and select category. After that, they can see the post thumbnail after filtering. By clicking into the post, users can view the whole post and see every detail of the post. Besides, they can like the post by clicking the button and add comment.

If the users are admin, they can login the system by clicking the login button. If they cannot pass the authentication verification, they can browse, like and comment the posts only. If the Admin login the system successfully, they can also add, edit, delete post directly by clicking different button. They can logout the system and view the post like normal users.

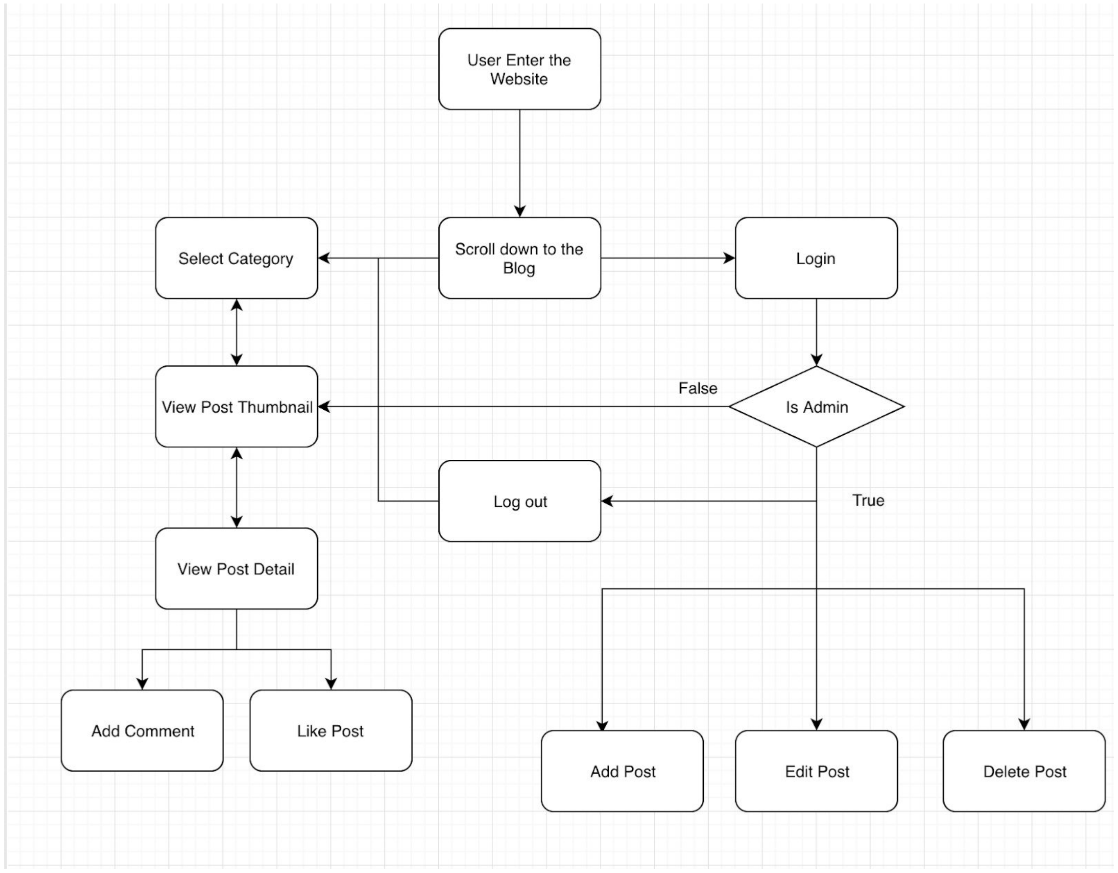


Figure 2. Flowchart

- **Data stream interaction**

Here is a flowchart specify that how our system interact with MVC component as well as database. For all users who enter our website, the controller will get post data from database to them by sending a trigger request to model. After getting data, the model renders the post data to the view so the user can see the posts in the website. When they click the category button, it processes in the similar way. The only different is the interaction with data model. For example, the category function filter will use the where() function provided by firestore API.

As for the admin, they can add or edit the post by sending a request and interact with the data model by sending the data from the view. then the model update the database. The delete is to send a request for deleting post with ID and update the database.

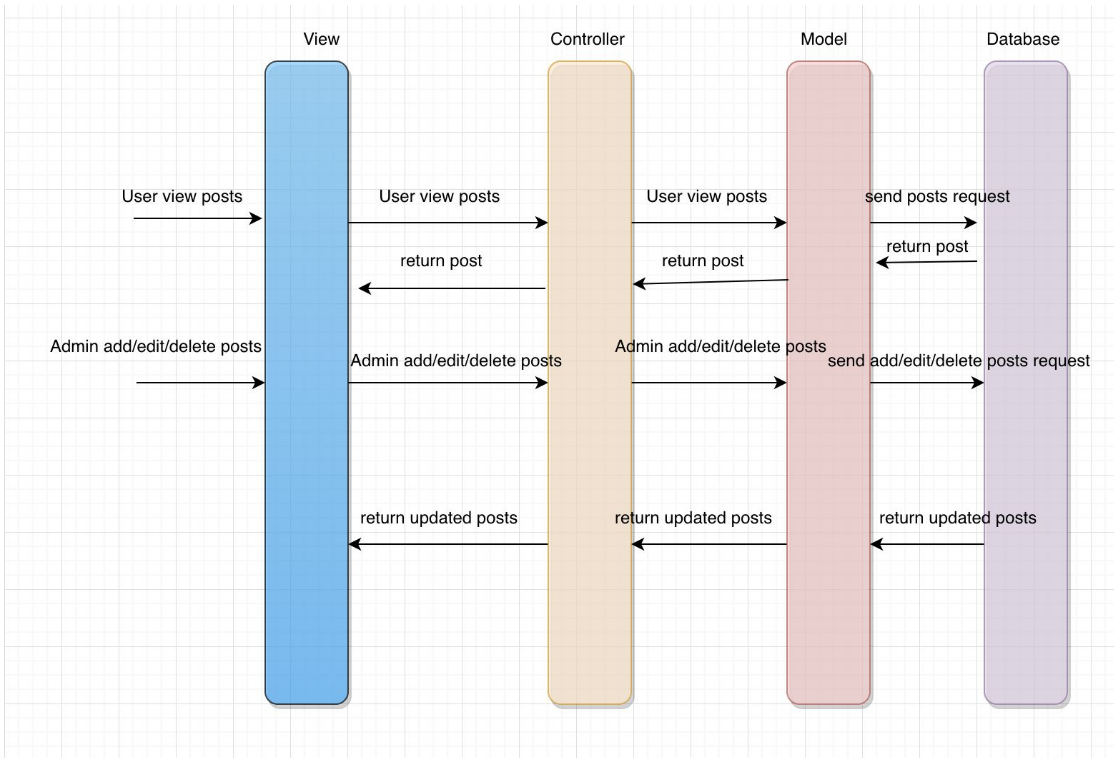
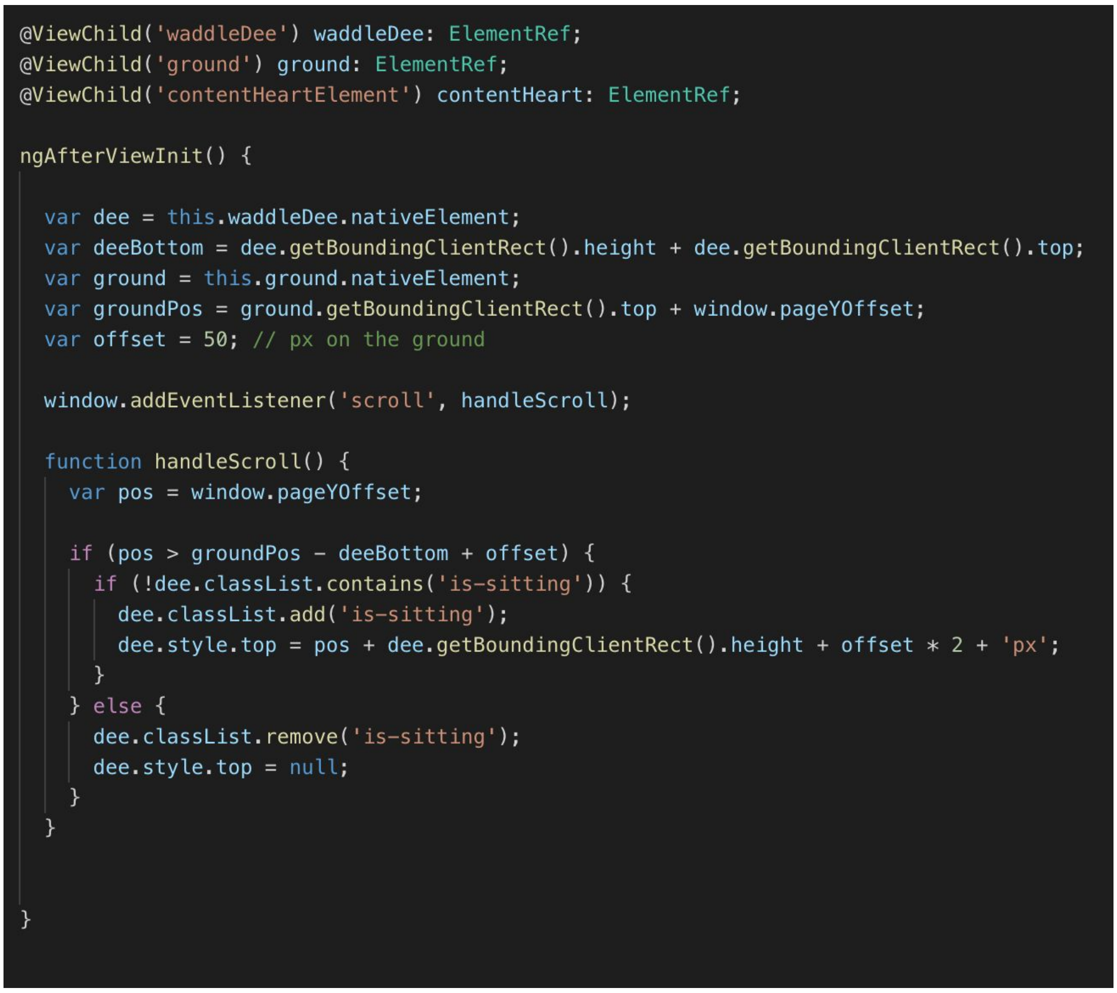


Figure 3. Data stream interaction

Components of the system & Application logic & Functionalities

- ### CSS & JavaScript animation

This website contains a CSS animation. After scrolling down, it will display the blog page. The animation effect is purely using CSS and Javascript. All elements such as cloud, character, parachute, sky, grass are designed by using CSS. And it is no image for the scene. For using the Angular, we here use TypeScript instead of JavaScript for these animation. Give these codes for example. we detect the Waddle Dee landing in the ground by detecting the position and the ground position, then add a class to Waddle Dee in order to make it sitting in the grass and waving his hands.



```
@ViewChild('waddleDee') waddleDee: ElementRef;
@ViewChild('ground') ground: ElementRef;
@ViewChild('contentHeartElement') contentHeart: ElementRef;

ngAfterViewInit() {

  var dee = this.waddleDee.nativeElement;
  var deeBottom = dee.getBoundingClientRect().height + dee.getBoundingClientRect().top;
  var ground = this.ground.nativeElement;
  var groundPos = ground.getBoundingClientRect().top + window.pageYOffset;
  var offset = 50; // px on the ground

  window.addEventListener('scroll', handleScroll);

  function handleScroll() {
    var pos = window.pageYOffset;

    if (pos > groundPos - deeBottom + offset) {
      if (!dee.classList.contains('is-sitting')) {
        dee.classList.add('is-sitting');
        dee.style.top = pos + dee.getBoundingClientRect().height + offset * 2 + 'px';
      }
    } else {
      dee.classList.remove('is-sitting');
      dee.style.top = null;
    }
  }


}
```

Figure 4. Implement Waddle Dee Animation in TypeScript

- **Blog body**

The blog system contains different modules and functionalities:

1. Login and logout
If it is not admin user, there is a login button. When user clicks the login button, it will jump to the Google account login page. If login success, the blog title will display user's blog name. Also, there is a logout button. Otherwise, it will display "you are not admin!".

All these are implemented by the "LoginComponent" in Angular. In this component, it will verify user's email. Only if it matches the account of Firebase database account, user can login.

2. Post categories
There are many categories and each category contains different posts. In "All" category, it will list all posts of the blog owner. When click the posts of other

categories, it will display the selected post.

These categories are dropdown buttons. When hovering the button, it will show different categories of the posts. If there is no post in the category, it will return “no result”.

3. Add post

There is an “Add Post” button. When the admin clicks this button, it will show the add post field. In our project, we support markdown format edit. Once click the “Add Post” button, the new post can be upload to cloud database.

4. Posts list

It will display all the posts here. It simply shows the main information of each post, such as title, body, comments and likes. When click the particular post, it will show the whole post.

5. View post

When open a post, it will display the post in markdown format. Each post contains a big post image, author, create date and the post body.

6. Edit/delete post

Only if the users login our system the edit/delete button will display. It will have a fade in and out effect when the mouse hovers the post. Click the “edit” icon button can open the markdown edit page. When save the post, it will upload to the cloud database. Click the “delete” icon button, it will delete this post from the cloud database.

7. Add comments

Under the comments, there is a “Add comment” field. Users can make comments of the post. All users can add comments whether he/she is an admin or not. When click the “Add a comment” button, the name as well as the comment will be sent to the cloud database and immediately displayed on the comments area under this post. In addition, our blog can display the number of comments. In the comment field, there are lists of comments with names and contents.

8. Add likes

Under the post, there are many comments of the post. And there is a “like” button. Other users can like the post even they don’t log in our blog system. When user clicks the like button, the likes count will add one and the heart icon will turn red. Besides, users can not press like button again.

Class diagrams

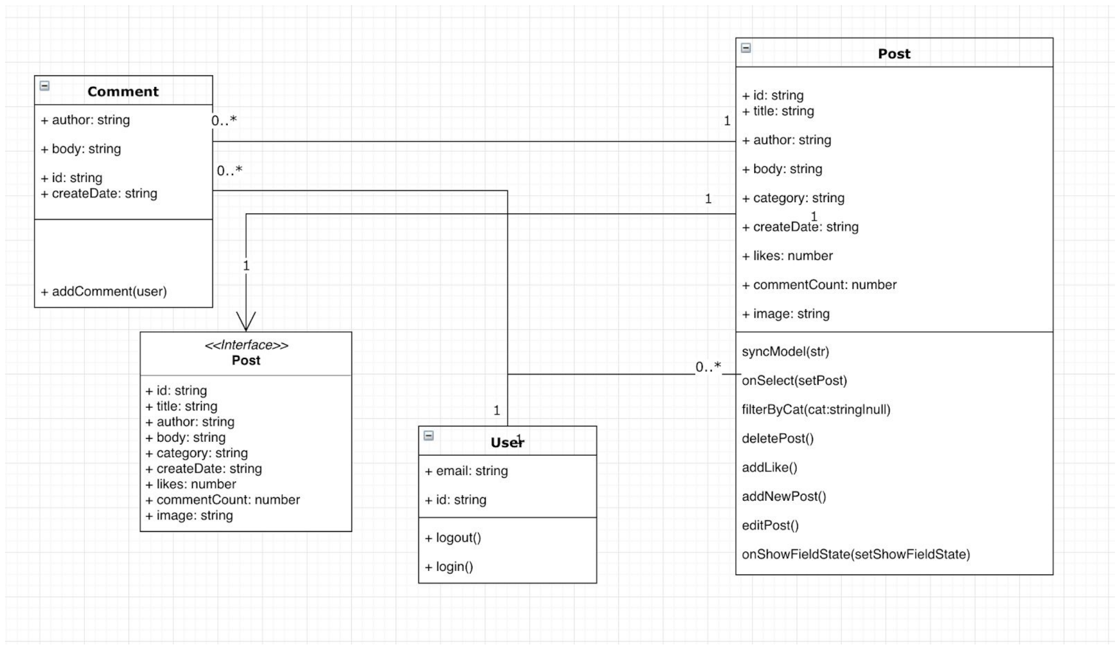


Figure 5. Class diagram

This is a class diagram of our database. It contain a Post. and It refer to the Interface <<Post>> and contain value of id, title, author, body, category, createDate, likes, commentCount, image. And have 8 method. the syncModel() is for the markdown editing, which bind the value of the text field to the markdown component and display. the onSelect() function determine which post are being selected and showed the post's detail. FilterByCat() function is for the category filtering. deletePost() function is for deleting post. addLike() is for adding a likes for a post. addNewPost() function is for adding new post. editPost() is for submit the edited post and update. onShowFieldState() function is for determine now what element should be showed such as post's thumbnail, post's detail, post's editor for adding new post, post's editor for editing post. the other class is quite simple to be understand. User is for the login component. only Admin can manage post. it contain some basic information, email and id. The comment is storing the comment. It is a subcollection in the document. Meaning that each comment is a document of a subcollection of a post's document. comment have one Post. Each post can have 0 to many comment. Each user have 0 to many comments, each comment have only 1 user. each user have 0 to many post. Each post has one user.

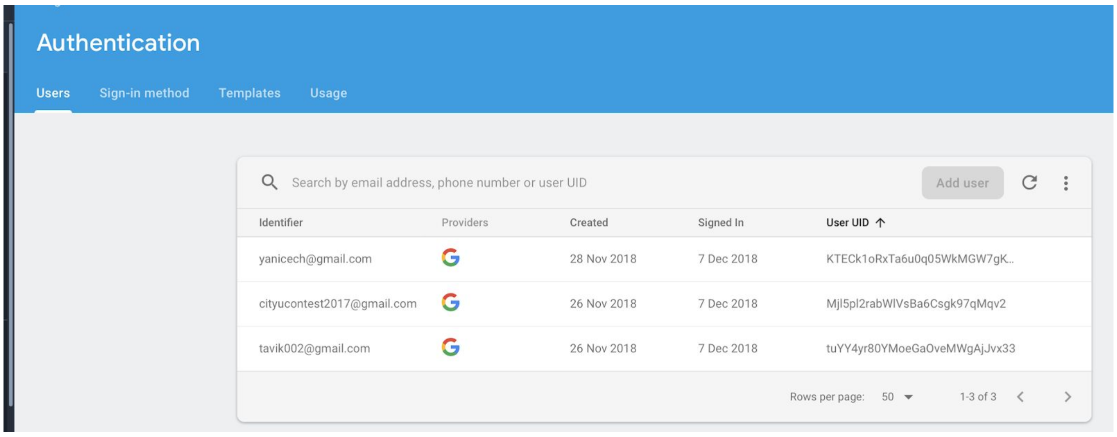
Database

In the blog system, we make use of Firestore provided by Firebase, Google. Firestore is a NoSQL database. We use NoSQL database because it is flexible and useful. We will not storing business critical data or financial data to the database. so RDBMS is not suitable to the blog system. The system does not require ACID.

Firestore is a cloud database meaning that we can connect to the database everywhere everytime without downloading database software such as mySQL, mongoDB and we do not need to create our own database in our computer and turn on the database server. Firestore is a document based database. All data are got in JSON format. and they are key-value pair.

In Firestore database, there is a collection, and the collection can have many documents. Each document can contain lots of files. Every file has a list of “key-value” pair of data.

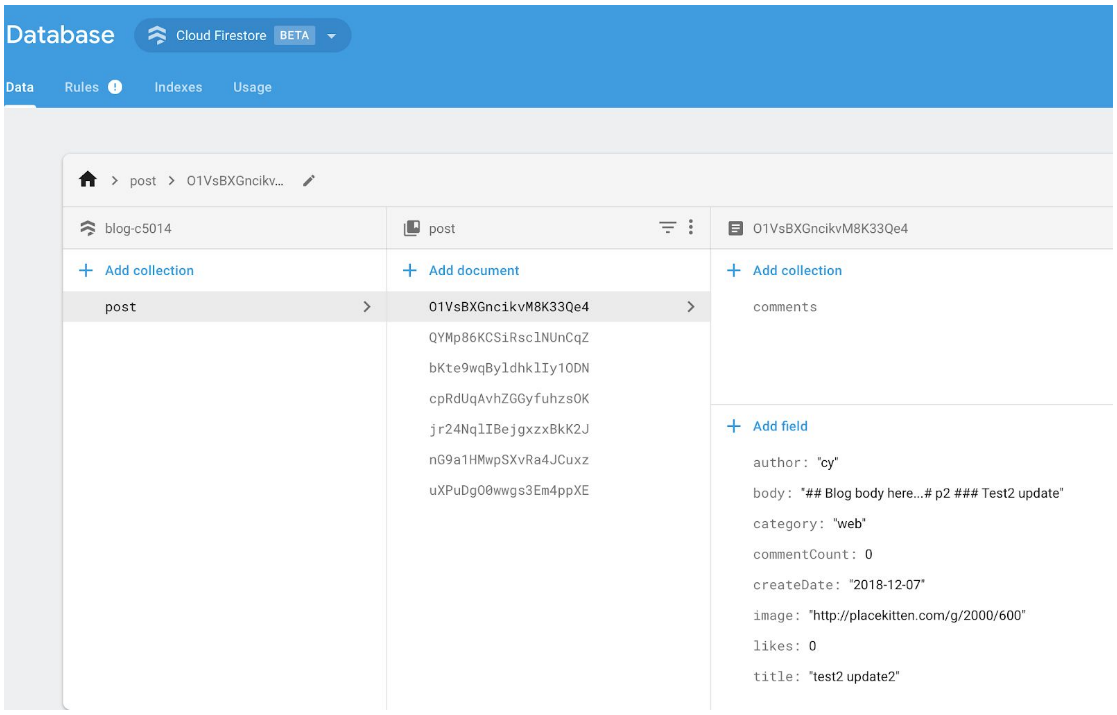
In our project, we have different users. And each user has one “Post” collection. Inside “Post” collection, there are many “post” documents. Each “post” has “title”, “author”, “body”, “category”, “createDate”, “likes”, “commentCount”, “image” and “comments”. and our comments is subcollection related to the post document. each post has it’s own comments subcollection.



Authentication

Identifier	Providers	Created	Signed In	User UID ↑
yanicech@gmail.com	G	28 Nov 2018	7 Dec 2018	KTECk1oRxTa6u0q05WkMGW7gK...
cityucontest2017@gmail.com	G	26 Nov 2018	7 Dec 2018	Mjl5pl2rabWlVsBa6Csgk97qMqv2
tavik002@gmail.com	G	26 Nov 2018	7 Dec 2018	tuYY4yr80YMoeGaOveMWgAjJvx33

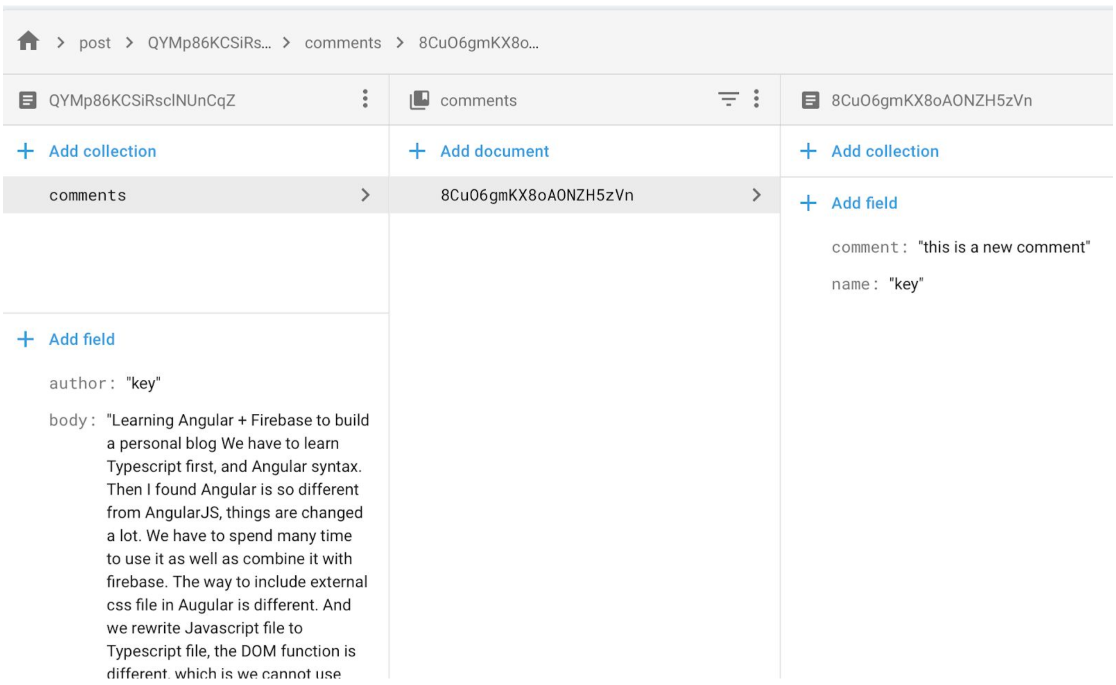
Rows per page: 50 1-3 of 3



01VsBXGncikvM8K33Qe4
QYMp86KCSiRsclNUnCqZ
bKte9wqByldhklIy1ODN
cpRdUqAvhZGGyfuhzsOK
jr24NqlIBejgxzxBkK2J
nG9a1HMwpSXvRa4JCuxz
uXPuDgO0wwgs3Em4ppXE

comments

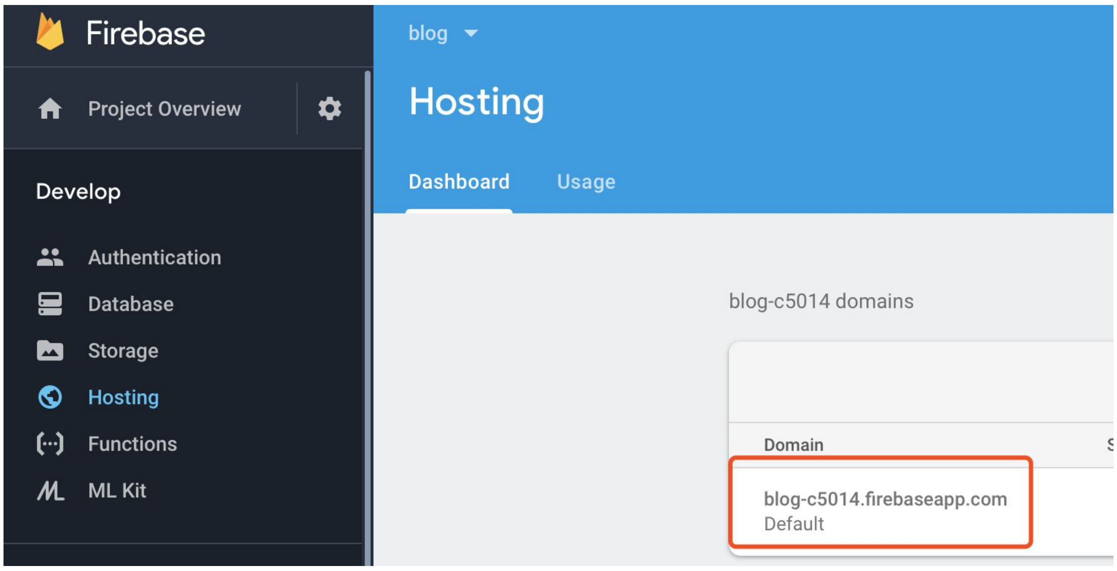
author: "cy"
body: "## Blog body here...# p2 ### Test2 update"
category: "web"
commentCount: 0
createDate: "2018-12-07"
image: "http://placekitten.com/g/2000/600"
likes: 0
title: "test2 update2"

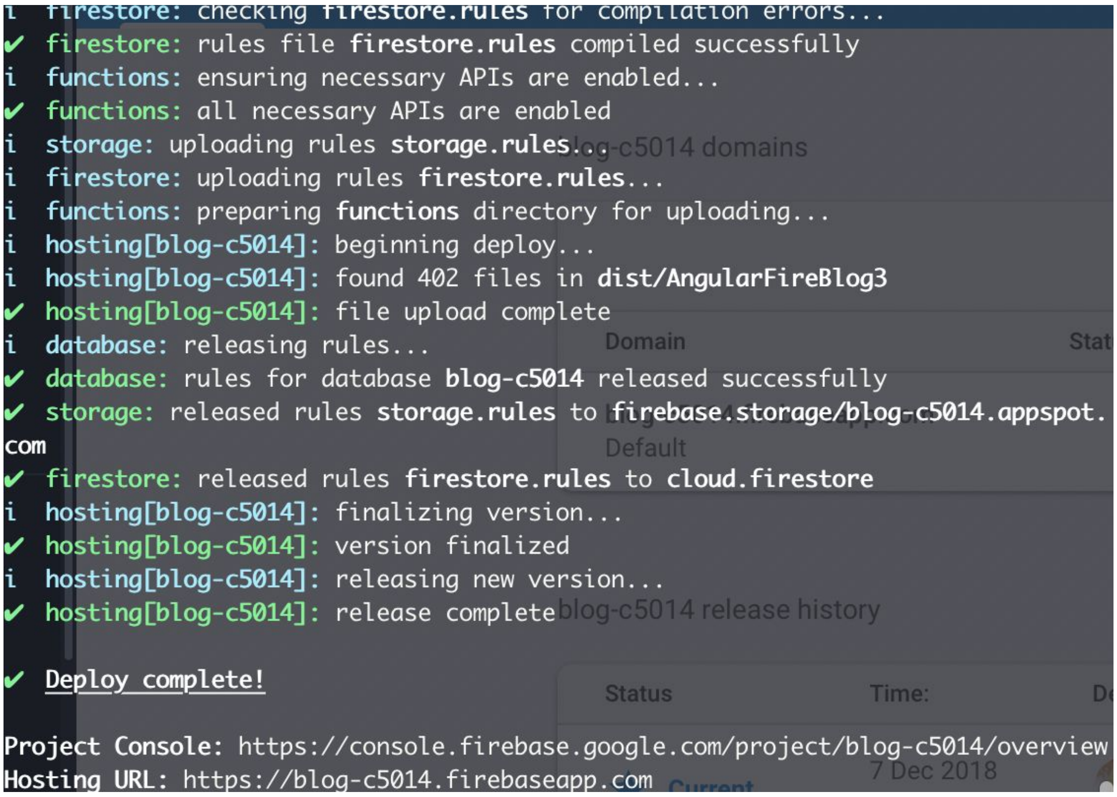


comment: "this is a new comment"
name: "key"

author: "key"
body: "Learning Angular + Firebase to build a personal blog We have to learn Typescript first, and Angular syntax. Then I found Angular is so different from AngularJS, things are changed a lot. We have to spend many time to use it as well as combine it with firebase. The way to include external css file in Augular is different. And we rewrite Javascript file to Typescript file, the DOM function is different, which is we cannot use

It should be noted that we have deployed our blog system to cloud hosting thanks to the hosting function provided by powerful Firebase.





```
i  firestore: checking firestore.rules for compilation errors...
✔  firestore: rules file firestore.rules compiled successfully
i  functions: ensuring necessary APIs are enabled...
✔  functions: all necessary APIs are enabled
i  storage: uploading rules storage.rules...
i  firestore: uploading rules firestore.rules...
i  functions: preparing functions directory for uploading...
i  hosting[blog-c5014]: beginning deploy...
i  hosting[blog-c5014]: found 402 files in dist/AngularFireBlog3
✔  hosting[blog-c5014]: file upload complete
i  database: releasing rules...
✔  database: rules for database blog-c5014 released successfully
✔  storage: released rules storage.rules to firebase.storage/blog-c5014.appspot.
com
✔  firestore: released rules firestore.rules to cloud.firestore
i  hosting[blog-c5014]: finalizing version...
✔  hosting[blog-c5014]: version finalized
i  hosting[blog-c5014]: releasing new version...
✔  hosting[blog-c5014]: release complete

✔  Deploy complete!

Project Console: https://console.firebase.google.com/project/blog-c5014/overview
Hosting URL: https://blog-c5014.firebaseapp.com
```

Programming (programming languages, tools, framework)

Programming languages: HTML, CSS, JavaScript, TypeScript

Framework: Bootstrap, AngularFire(Angular + Firebase)

Angular6 + Firebase
We use Angular6 + Firebase in this project. There are many components in the website, for example, app component, login component, editor component and other components. Different components contain three main files: HTML, CSS and ts. In HTML, we write the HTML code of the website. In CSS, we write the CSS code. And in ts, we write the data manipulation code here.

The biggest advantage of Angular and Firebase is that we can bind the data between front-end and cloud database together using TypeScript. In the front-end we can display the data in cloud database easily and in the back-end we can get the data from the webpage once the data input has changed.

With Angular6 + Firebase framework, we can better implement our blog system in MVC architecture. Using the component in Angular, we can develop the layout of our blog system in the component.html and component.css. We can define and specify our data manipulation functions within the component.ts. Also, we can manage our data in the Cloud Firestore database.

We store all the blog posts data in the Cloud Firestore database. It is NoSQL database and it provides real-time data manipulation, which helps us process flexible data more easily.

We can design a comprehensive web app thanks to Angular. A popular feature of Angular is the single view app design making it possible to display different content without reloading the page. The most frequency for single web design is making use of '*ngIf ' function provided by Angular Cli

Angular cli support TypeScript with is a strong language compare to weak language JavaScript. The variable should be assigned a data type in TypeScript. And in TypeScript, we can use import {} from '' so that we can use many different functions provided by others. TypeScript is stricter than the 'use strict' mode of JavaScript making debugging easier. And developers can follow the standard of TypeScript coding and the coding could be more readable than JavaScript.

Testing strategies & Result:

For the testing, we apply TDD (Test driven development) and Pair Programming in the project. we first specify and define the test case of each function and then develop it and pass the test case. We suppose the expected output and compare it to the real output. And if it's not identical, then we use debugging knowledge base on control variates to find out the reason for the problem.

For example:

Test case 1: Click into thumbnail and view post detail

Result 1: The results is it shows the detail of the post include title, time, author, category, and body and comments in the post. The test is passed.

Test Case 2: CRUD Management

Result 2: After login the system, admin can use the add new post, update post, delete post smoothly. And the database updates the data automatically. The test is passed.

We use these test strategies for all testing from the main function to every detail of the blog system.

Difficulties & Solutions

- **HTML & CSS layout**

Difficulties:

- Waddle Dee animation & layout reorganize
- Button layout design
- Post block position
- Edit/Delete fade in and fade out effect
- Grass & Ground CSS layout bug

Solutions:

- Use Bootstrap
- Sometime not to use bootstrap
- Debug the elements in the browser and fix it

- **Front-end & Back-end interaction**

Difficulties:

- Failure in connecting front-end and back-end
- Beginner of Angular6, Firebase, and TypeScript with limited knowledge of this framework
- Difficulties in data transfer between different components when using Angular6
- Difficulties in data manipulation between Angular6 and Firestore (Add/Edit/Delete operation)
- Difficulties in supporting markdown format editor and markdown format display
- Difficulty in passing user login information after login verification.
- Transfer from AngularJS to Angular Cli

Solutions:

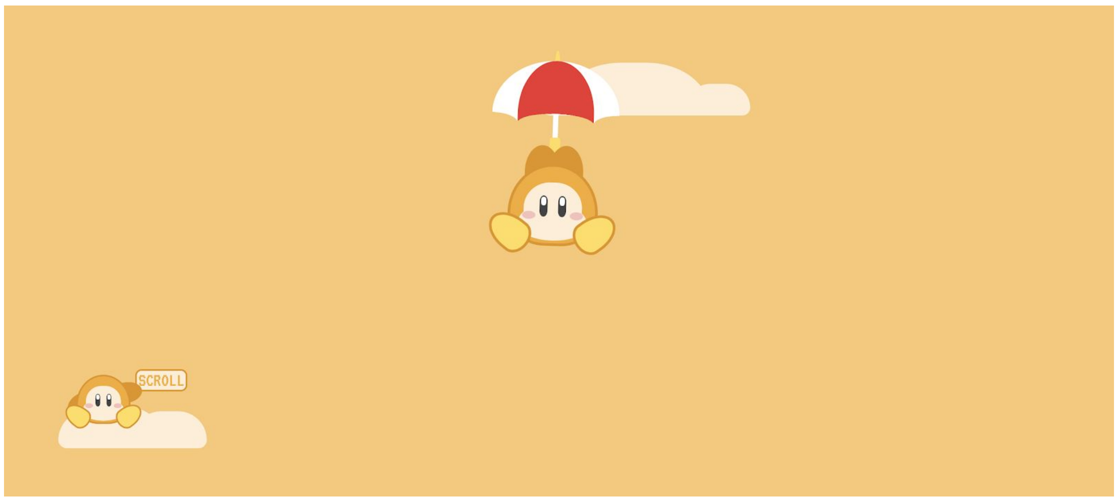
- Use Angular6 + Firebase framework
- Google their official documents, Github, StackOverflow, and other related blogs and watch videos on Youtube.
- Figure out two-way data binding framework in Angular6 (ngModel, Parent-Child component interaction).
- Use Angular-supported editor.md Markdown editor.

Evolution (Future Work)

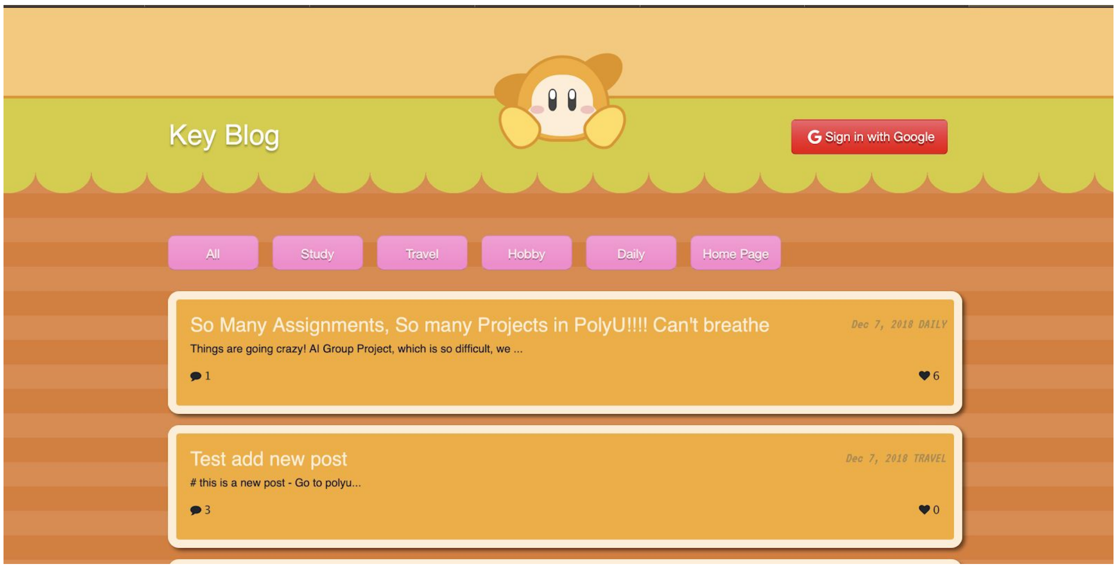
- Duplicate the blog and make a blog for Chen Yongxin also.
- Image upload problem when using Markdown editor.
- Add comment create date & time.
- Fix the white space on the right issue for small devices when developing RWD.
- Add function for uploading an image to firestore.
- Add a function for checking authentication when updating and log out at this moment.
- Add a function for the user to login account for post comment.
- Detect each user can like a post only one time.
- Design a confirmation message for popping up when clicking the delete button.
- Fix the markdown element showed in the thumbnail problem.
- Fix Waddle Dee's animation bugs for landing relating to Typescript.
- Fix the bug about updating image URL error when updating the post.
- Refactoring the code and make it clear for MVC design.
- Clear the test data in the database and organize the data.
- Add page number navigation and each page displays 10 posts at most.

User manual

i. Scrolling down when loaded the web page.



ii. Browse the posts of the blog when waddle dee landed in the ground



So Many Assignments, So many Projects in PolyU!!!! Can't breathe

Dec 7, 2018 DAILY

Things are going crazy! AI Group Project, which is so difficult, we ...

1 6

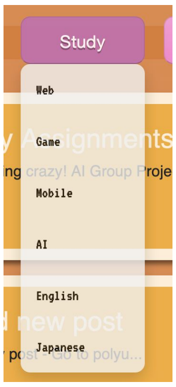
Test add new post

Dec 7, 2018 TRAVEL

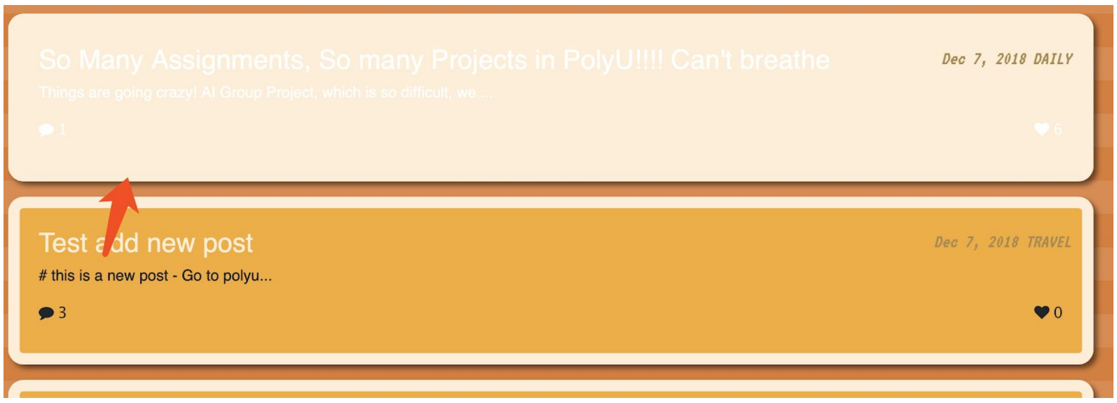
this is a new post - Go to polyu...

3 0

iii. Select category to filter the posts



iv. View the post by clicking into the post



So Many Assignments, So many Projects in PolyU!!!! Can't breathe

Dec 7, 2018 DAILY

Things are going crazy! AI Group Project, which is so difficult, we ...

1 6

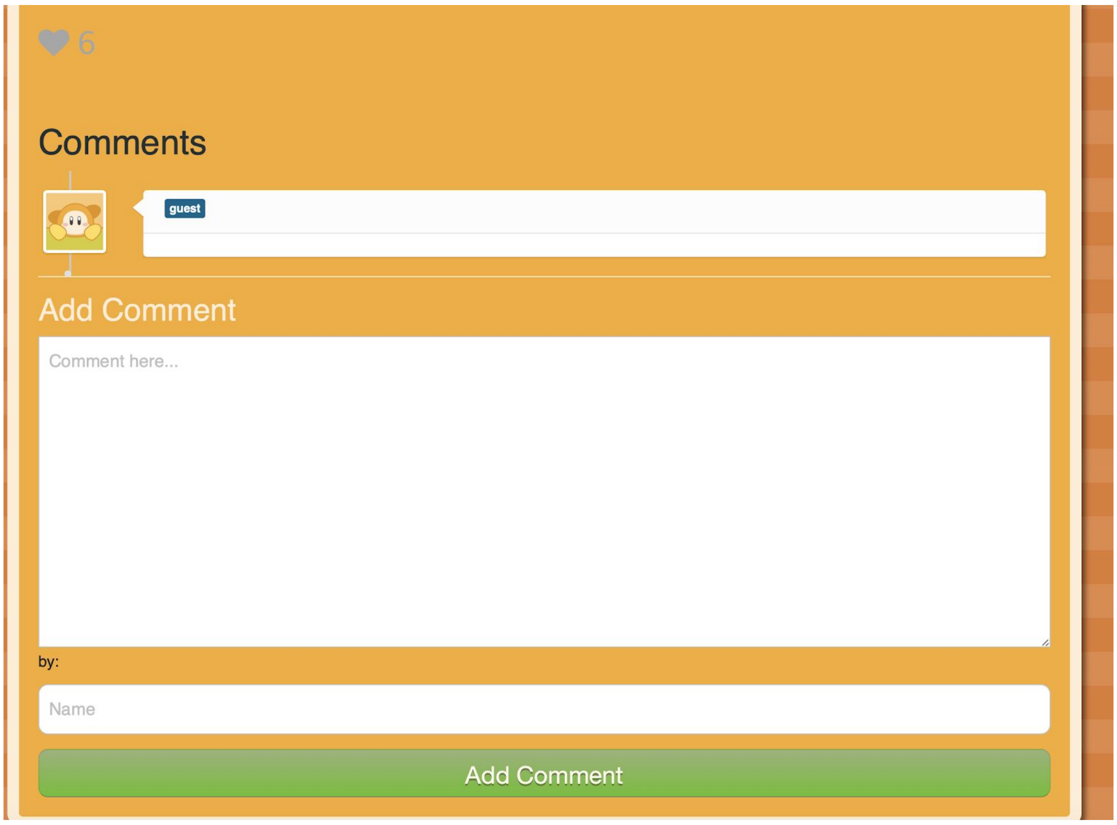
Test add new post

Dec 7, 2018 TRAVEL

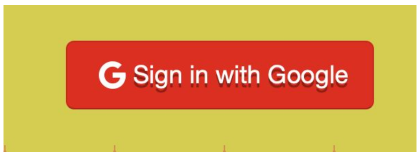
this is a new post - Go to polyu...

3 0

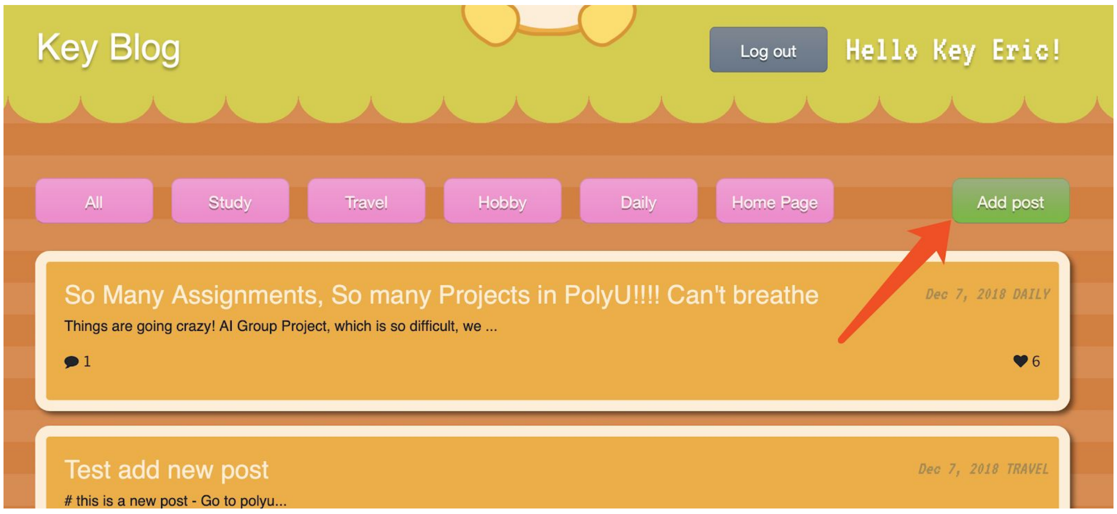
v. like the post by clicking the heart icon. view the comment, add a comment by fill into the comment text holder and press add comment button to submit a comment



vi. click category 'All' to back to post's thumbnail page.
vii. Login with Google



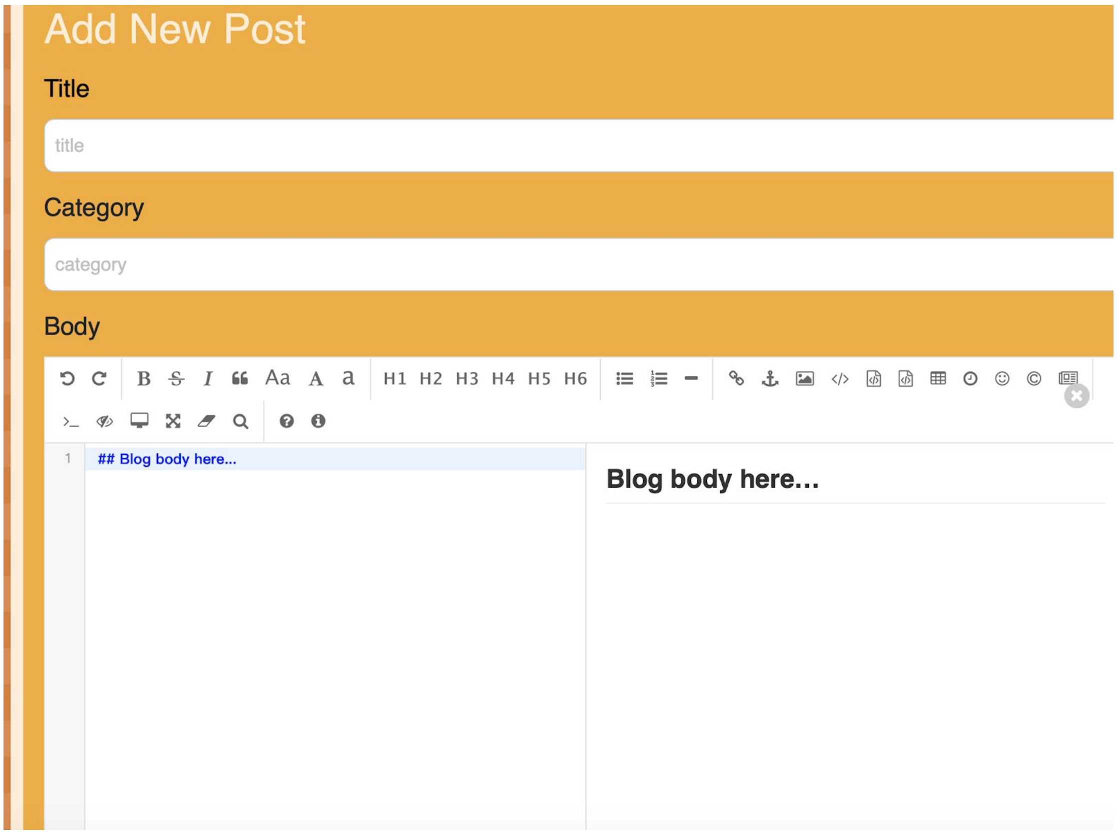
viii. If you are the admin. then you can see the add post button. click it for adding a post



So Many Assignments, So many Projects in PolyU!!!! Can't breathe
Dec 7, 2018 DAILY
Things are going crazy! AI Group Project, which is so difficult, we ...
1
6

Test add new post
Dec 7, 2018 TRAVEL
# this is a new post - Go to polyu...

ix. type all content of the posts. then click Add post button to submit, cancel button to cancel.



Add New Post

Title

title

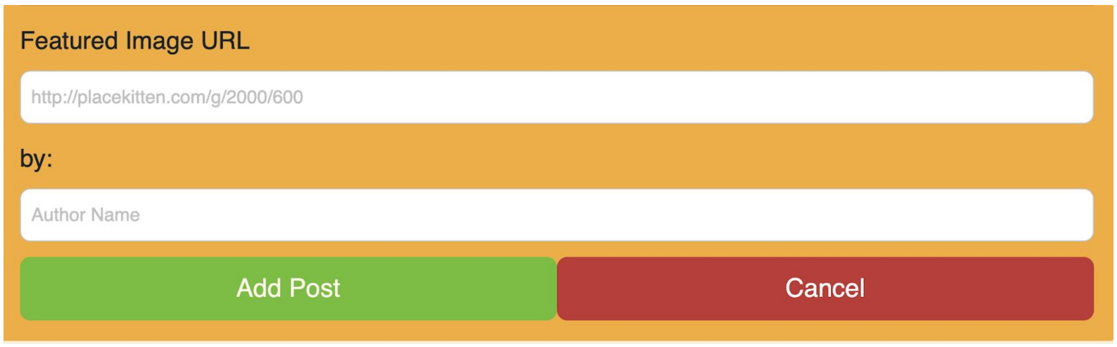
Category

category

Body

```
## Blog body here...
```

Blog body here...



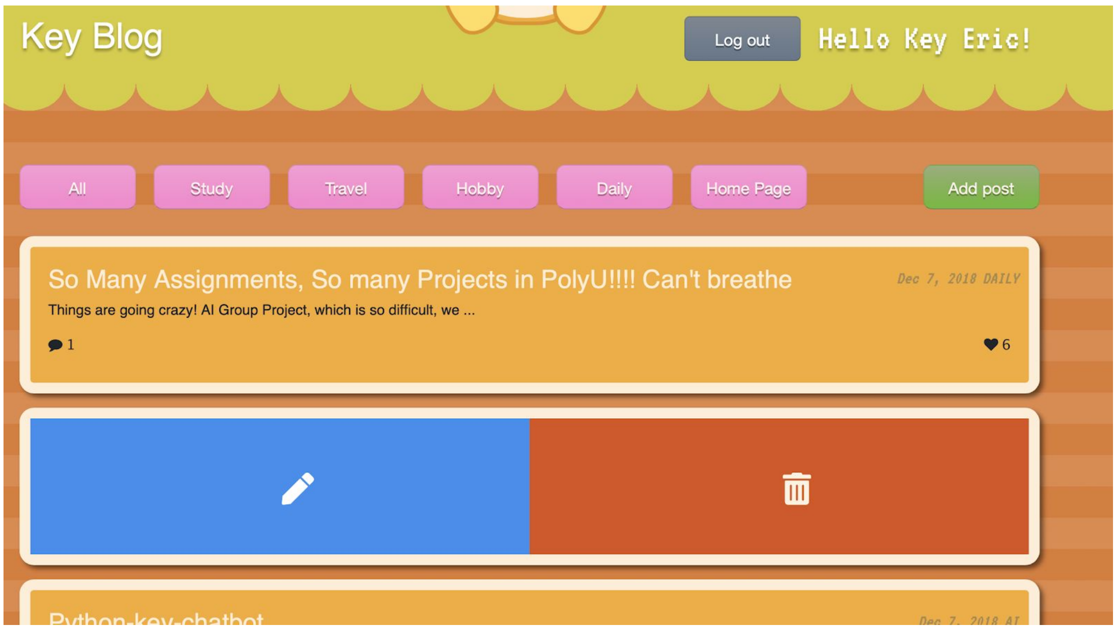
Featured Image URL

http://placekitten.com/g/2000/600

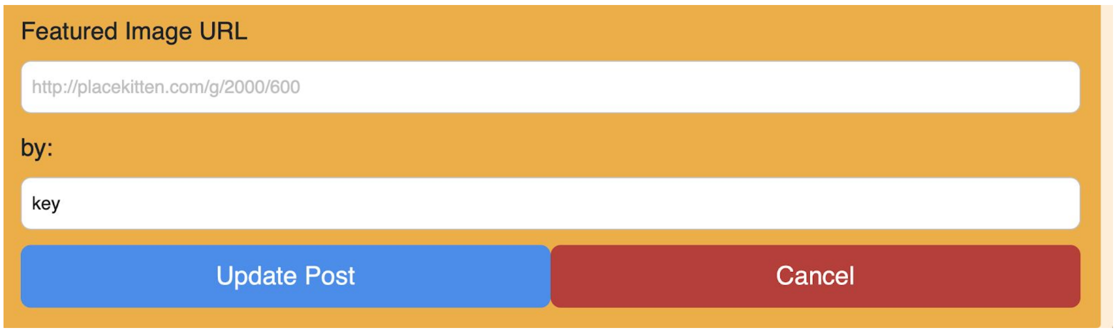
by:

Author Name

x. if you are the admin, you can hover the post to show the edit and delete button


Key Blog
Log out
Hello Key Eric!
All
Study
Travel
Hobby
Daily
Home Page
Add post
So Many Assignments, So many Projects in PolyU!!!! Can't breathe
Dec 7, 2018 DAILY
Things are going crazy! AI Group Project, which is so difficult, we ...
1
6

xi. click delete to delete a post. click edit for editing. editing is same with add post. but the submit button is updated.



xii. click log out button for log out

Installation guide

For the installation guide, please refer to the readme.md file or check my Github repos for information. https://github.com/tavik000/AngularFireBlog#contents

Role and contributions of group members

Name	Role	Contribution
Zhao Haiqi	Full Stack Developer: 1.Angular Research and development 2. Firebase Research and development 3. CSS HTML development 4. MVC design 5. RWD design 6. Database design 7. Testing And Debugging 8. Knowledge Sharing 9.Animation Development ...	50%
Chen Yongxin	Full Stack Developer: 1. Angular Research and development 2. Firebase Research and development 3. CSS and HTML development 4. RWD testing 5. Testing and Debugging 6. Markdown editor plugin development 7. Knowledge Sharing ...	50%