

Lab 15 Posts (v 1.0)

User List

- bill
- steve

Posts : Bill Gates (1)

New OS : Windows 11

comments (2)

Name : Mark *Stars :* 4 *Date :* Wed Nov 30 2022 08:30:00 GMT+0700 (Indochina Time)

Cheer!

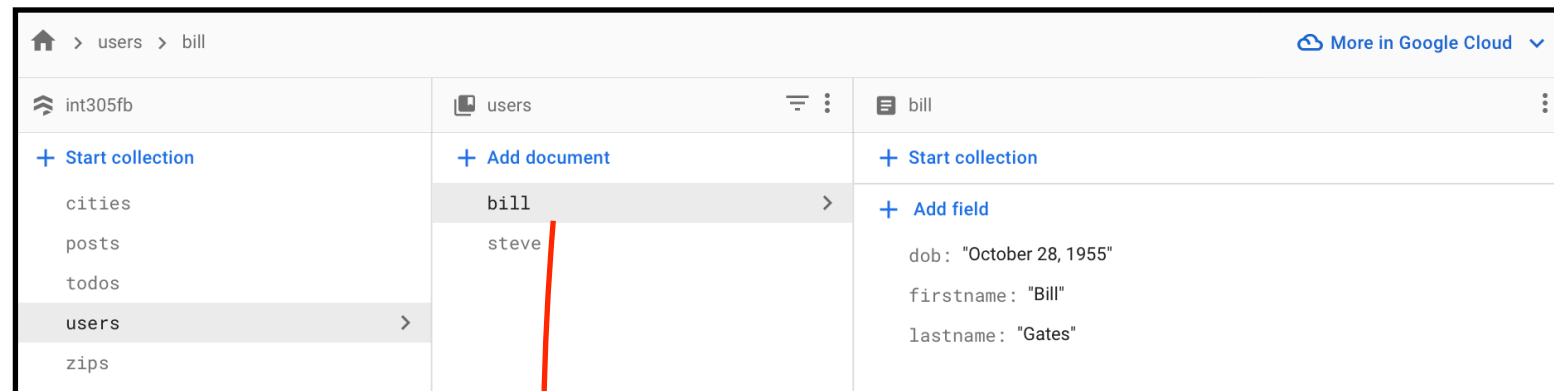
Name : Elon Musk *Stars :* *Date :* Wed Nov 30 2022 11:02:10 GMT+0700 (Indochina Time)

Release Date?

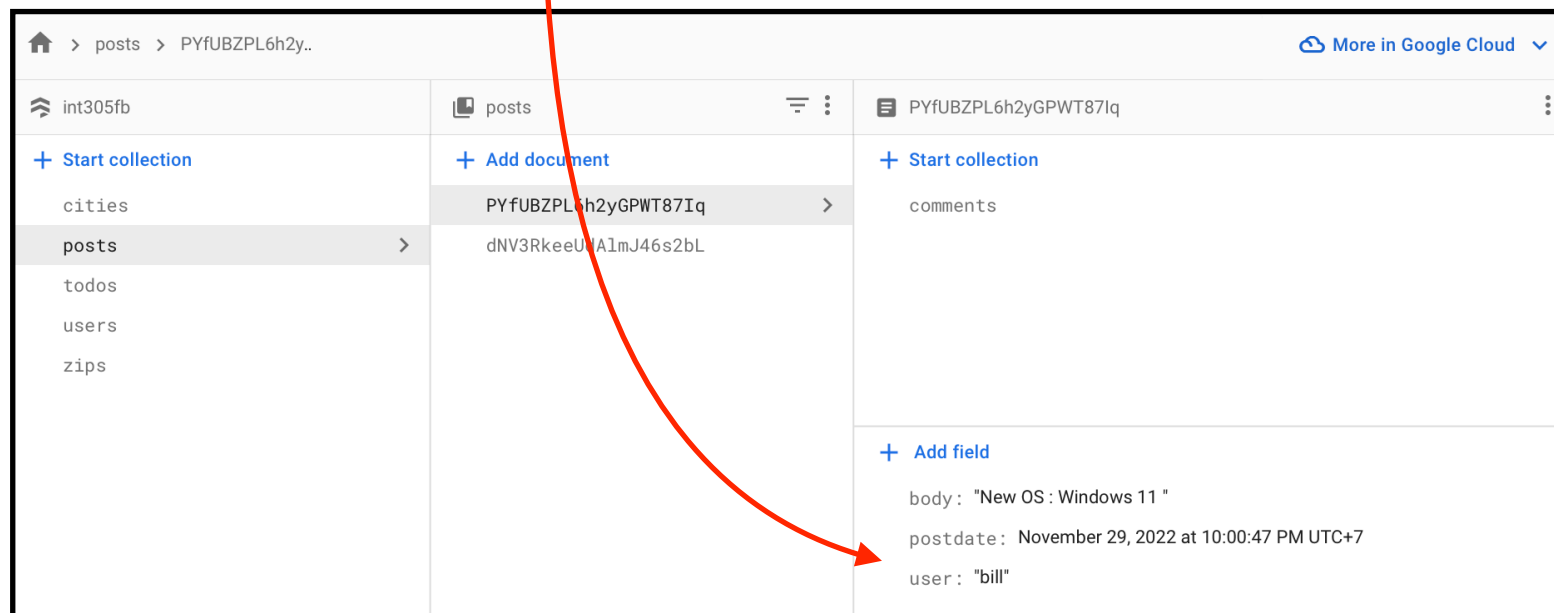
- users
 - id (doc id)
 - firstname string
 - lastname string
 - dob timestamp
- posts
 - id (doc id)
 - user string (user id)
 - postdate timestamp
 - body string
 - comments subcollection
 - name string
 - cmtdate timestamp
 - comment string
 - stars number

collections

users



posts

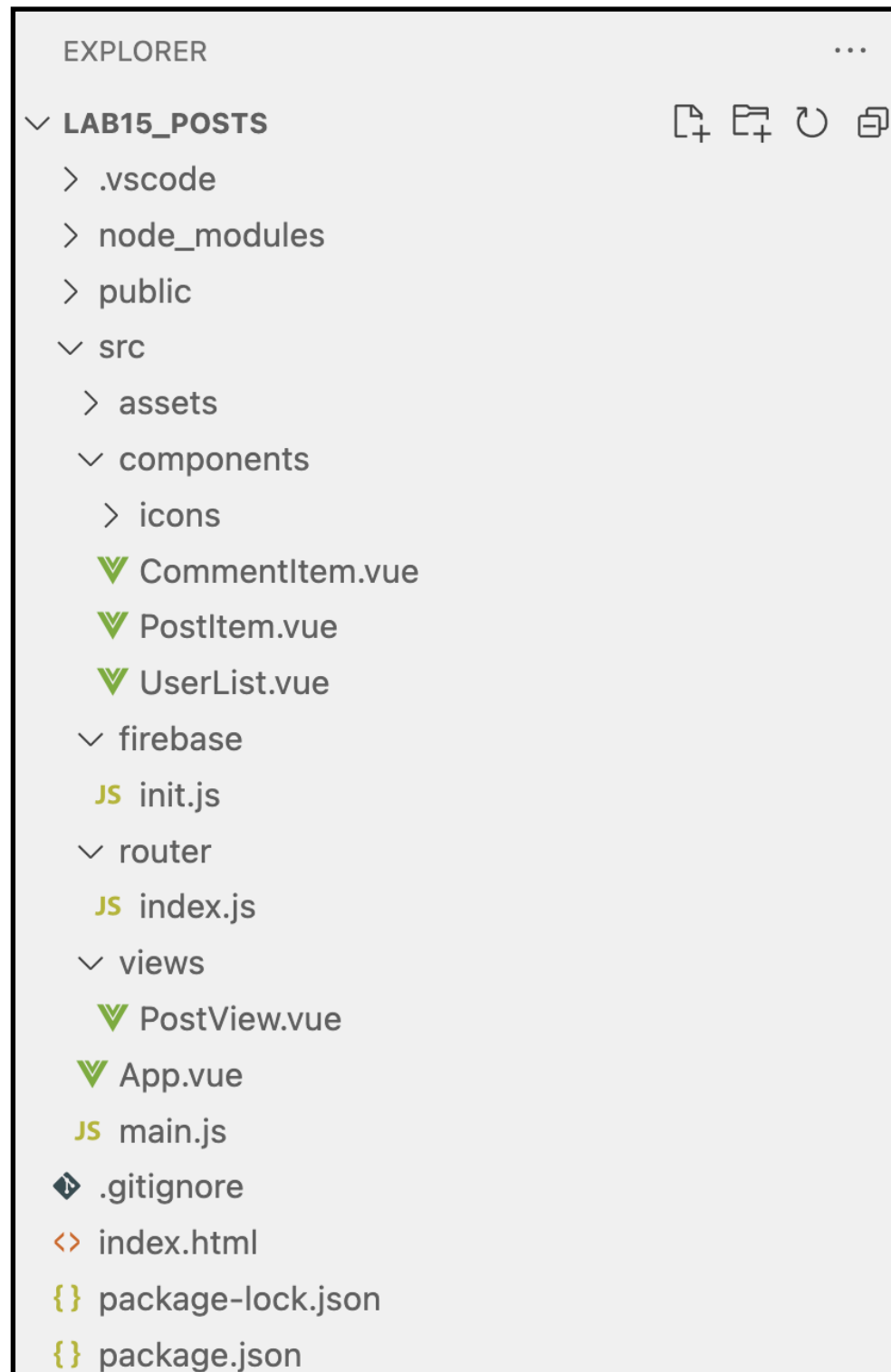


sub collection

comments

🏠 > posts > PYfUBZPL6h2y... > comments > 0AS7h9P0gp8... ☁ More in Google Cloud v		
☰ PYfUBZPL6h2yGPWT87lq ⋮	📄 comments ≡ ⋮	☰ 0AS7h9P0gp8WHmmOLVHl ⋮
+ Start collection	+ Add document	+ Start collection
comments >	0AS7h9P0gp8WHmmOLVHl > ln5QJbikR42VMPG6Y048	+ Add field cmtdate: November 30, 2022 at 8:30:00 AM UTC+7 comment: "Cheer!" email: "mark@fb.com" name: "Mark" stars: 4
+ Add field body: "New OS : Windows 11 " postdate: November 29, 2022 at 10:00:47 PM UTC+7 user: "bill"		

Project Structure



- `cd lab15_posts`
- `npm install`

src/firebase/init.js

```
JS init.js ×
src > firebase > JS init.js > ...
1  // Import the functions you need from the SDKs you need
2  import { initializeApp } from "firebase/app";
3  // TODO: Add SDKs for Firebase products that you want to use
4  // https://firebase.google.com/docs/web/setup#available-libraries
5  import { getFirestore } from "firebase/firestore" ;
6
7  // Your web app's Firebase configuration
8  const firebaseConfig = {
9    apiKey: "",
10   authDomain: "",
11   projectId: "",
12   storageBucket: "",
13   messagingSenderId: "",
14   appId: ""
15 };
16
17
18 // Initialize Firebase
19 const app = initializeApp(firebaseConfig);
20
21 // init firestore service
22 const db = getFirestore()
23
24 export default db
```

src/router/index.js

```
JS index.js  ×
src > router > JS index.js > ...
1  import { createRouter, createWebHistory } from 'vue-router'
2  import PostView from '../views/PostView.vue'
3
4  const router = createRouter({
5    history: createWebHistory(import.meta.env.BASE_URL),
6    routes: [
7      {
8        path: '/posts/:user',
9        name: 'posts',
10       component: PostView
11     }
12   ]
13 })
14
15 export default router
16
```

src/App.vue

```
App.vue ×
src > App.vue
1  <script setup>
2  import { RouterLink, RouterView } from 'vue-router'
3  import { ref, onMounted } from 'vue'
4  import { query, collection, getDocs } from 'firebase/firestore'
5  import db from './firebase/init.js'
6  import UserList from './components/UserList.vue'
7
8  const users = ref([])
9
10 async function getUsers(){
11   /* add your code here */
12 }
13
14 onMounted(() => {
15   getUsers()
16 })
17 </script>
18
19 <template>
20   <div>
21     <div>
22       <UserList :users="users" />
23     </div>
24   </div>
25   <div class="content">
26     <RouterView />
27   </div>
28 </template>
29
30 <style scoped>
31
32 </style>
33
```

src/views/PostView.vue

```
▼ PostView.vue ×
src > views > ▼ PostView.vue
1  <script setup>
2  import { ref, onMounted, watch } from 'vue'
3  import { useRoute } from "vue-router"
4  import { collection, query, where, getDocs, onSnapshot } from "firebase/firestore"
5  import db from "../firebase/init.js"
6  import PostItem from "../components/PostItem.vue"
7
8  const user = ref("")
9  const posts = ref([])
10 const route = useRoute()
11
12 async function getPosts(){
13   user.value = route.params.user
14   /* add your code here */
15 }
16
17 watch( () => route.params.user, getPosts)
18
19 onMounted(() => {
20   getPosts()
21 })
22
23 </script>
24
25 <template>
26   <h3>Posts : {{user}}</h3>
27   <PostItem v-for="post in posts" :post="post" :key="post.id" />
28 </template>
29
30 <style scoped>
31
32 </style>
33
```


src/components/UserList.vue

```
▼ UserList.vue ×
src > components > ▼ UserList.vue
1  <script setup>
2  import { RouterLink, RouterView } from 'vue-router'
3
4  defineProps({
5    users: {
6      type: Array,
7      required: true
8    }
9  })
10 </script>
11
12 <template>
13   <div class="greetings">
14     <h3>User List</h3>
15     <ul>
16       <li v-for="user in users" :key="user.id">
17         <RouterLink :to="'/posts/${user.id}'">{{user.id}}</RouterLink>
18       </li>
19     </ul>
20   </div>
21 </template>
22
23 <style scoped>
24
25 </style>
```

src/components/PostItem.vue

```
▼ PostItem.vue ×
src > components > ▼ PostItem.vue
1  <script setup>
2  import CommentItem from "../components/CommentItem.vue"
3
4  defineProps({
5    post: {
6      type: Object,
7      required: true,
8    }
9  });
10 </script>
11
12 <template>
13   <div class="box">
14     {{post.body}}
15     <h4 class="title">comments ({{post.comments.length}})</h4>
16     <CommentItem v-for="comment in post.comments" :comment="comment" :key="comment.id" />
17   </div>
18 </template>
19
20 <style scoped>
21   .box {
22     border: 1px solid grey;
23     padding: 5px ;
24     margin: 5px ;
25     border-radius: 5px ;
26   }
27   .title {
28     font-style: italic ;
29     color: hsla(160, 100%, 37%, 1);
30   }
31 </style>
```

src/components/CommentItem.vue

```
CommentItem.vue x
src > components > CommentItem.vue
1  <script setup>
2  defineProps({
3    comment: {
4      type: Object,
5      required: true,
6    }
7  });
8  </script>
9
10 <template>
11   <div class="comment">
12     <p class="commenter">
13       <span class="label">Name : </span><span>{{comment.name}}</span>
14       <span class="label">Stars : </span><span>{{comment.stars}}</span>
15       <span class="label">Date : </span><span>{{comment.cmtdate.toDate()}}</span>
16     </p>
17     <p>{{comment.comment}}</p>
18   </div>
19 </template>
20
21 <style scoped>
22   .comment {
23     margin-left: 10px ;
24     padding: 5px ;
25   }
26   .commenter {
27     display: grid ;
28     grid-template-columns: 1fr 1fr 1fr 1fr 1fr 5fr ;
29     font-size: smaller ;
30     font-style: italic ;
31   }
32   .content {
33     background-color: rgb(100,100,100) ;
34   }
35   .label {
36     color: hsla(160, 100%, 37%, 1);
37   }
```

Lab Tasks

1. Show user's name (firstname, lastname)
2. Application should reload when a new user is added.
3. Modify `getPosts()` to load the subcollection "comments"

```
const commentRef = collection(db, "posts", doc.id, "comments")
```

4. Show the number of posts for each user by applying Computed Pattern
5. Show the number of comments for each post by applying Computed Pattern
6. Use Extended Reference Pattern for the user name filed in the posts collection.
7. Add a component "NewComment" that allows users can write a new comment and the page should reload to show the new comment.