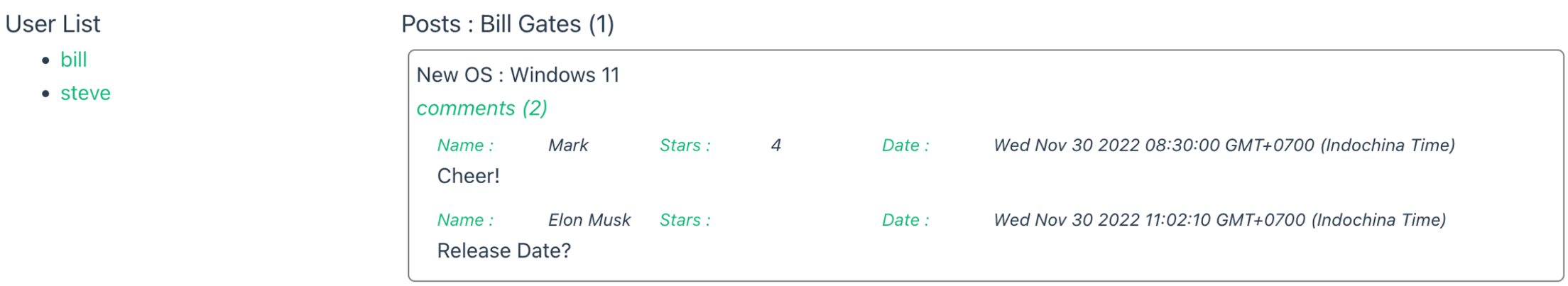
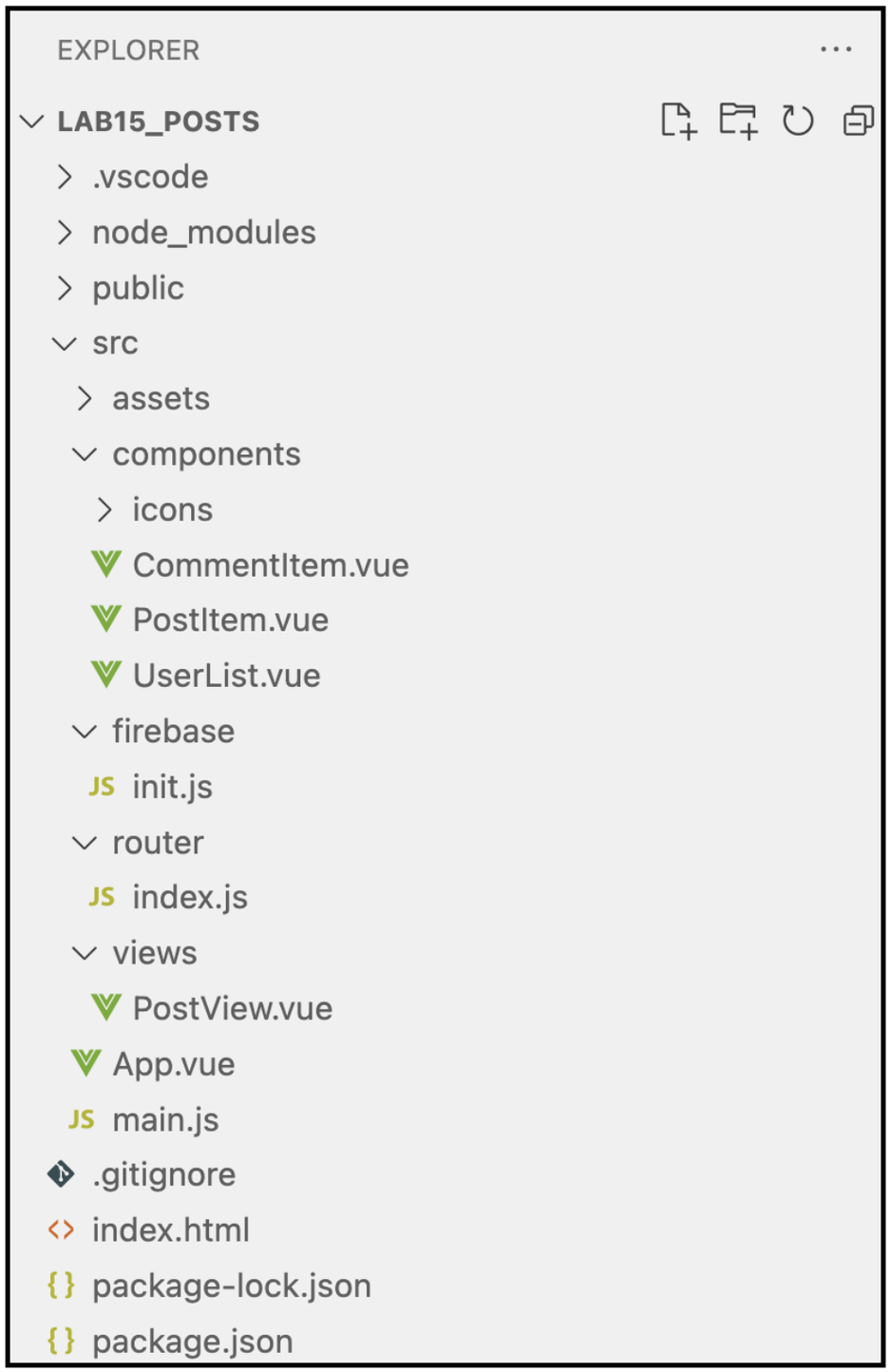
Lab 15 Posts (v 1.0)



- 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

Project Structure



- cd lab15_posts
- npm install

src/firebase/init.js

init.js

src > firebase > 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

index.js

src > router > 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

```vue
<script setup>
import { RouterLink, RouterView } from 'vue-router'
import { ref, onMounted } from 'vue'
import { query, collection, getDocs } from 'firebase/firestore'
import db from './firebase/init.js'
import UserList from './components/UserList.vue'

const users = ref([])

async function getUsers(){
  /*  add your code here */
}

onMounted(() => {
  getUsers()
})
</script>

<template>
  <div>
    <div>
      <UserList :users="users" />
    </div>
  </div>
  <div class="content">
    <RouterView />
  </div>
</template>

<style scoped>

</style>
```

src/views/PostView.vue

PostView.vue

src > views > PostView.vue

```vue
<script setup>
import { ref, onMounted, watch } from 'vue'
import { useRoute } from "vue-router"
import { collection, query, where, getDocs, onSnapshot } from "firebase/firestore"
import db from "../firebase/init.js"
import PostItem from "../components/PostItem.vue"

const user = ref("")
const posts = ref([])
const route = useRoute()

async function getPosts(){
  user.value = route.params.user
  /*  add your code here */
}

watch( () => route.params.user, getPosts)

onMounted(() => {
  getPosts()
})

</script>

<template>
    <h3>Posts : {{user}}</h3>
    <PostItem v-for="post in posts" :post="post" :key="post.id" />
</template>

<style scoped>

</style>
```

src/components/UserList.vue

UserList.vue

src > components > UserList.vue

```vue
<script setup>
import { RouterLink, RouterView } from 'vue-router'

defineProps({
  users: {
    type: Array,
    required: true
  }
})
</script>

<template>
  <div class="greetings">
    <h3>User List</h3>
    <ul>
      <li v-for="user in users" :key="user.id">
        <RouterLink :to="`/posts/${user.id}`">{{user.id}}</RouterLink>
      </li>
    </ul>
  </div>
</template>

<style scoped>

</style>
```

src/components/PostItem.vue

PostItem.vue

src > components > PostItem.vue

```vue
<script setup>
import CommentItem from "../components/CommentItem.vue"

defineProps({
  post: {
    type: Object,
    required: true,
  }
});
</script>

<template>
  <div class="box">
    {{post.body}}
    <h4 class="title">comments ({{post.comments.length}})</h4>
    <CommentItem v-for="comment in post.comments" :comment="comment" :key="comment.id" />
  </div>
</template>

<style scoped>
  .box {
    border: 1px solid grey;
    padding: 5px ;
    margin:  5px ;
    border-radius: 5px ;
  }
  .title {
    font-style: italic ;
    color: hsla(160, 100%, 37%, 1);
  }
</style>
```

src/components/CommentItem.vue

CommentItem.vue

src > components > CommentItem.vue

```vue
<script setup>
defineProps({
  comment: {
    type: Object,
    required: true,
  }
});
</script>

<template>
    <div class="comment">
      <p class="commenter">
        <span class="label">Name : </span><span>{{comment.name}}</span>
        <span class="label">Stars : </span><span>{{comment.stars}}</span>
        <span class="label">Date : </span><span>{{comment.cmtdate.toDate()}}</span>
      </p>
      <p>{{comment.comment}}</p>
    </div>
</template>

<style scoped>
  .comment {
    margin-left: 10px ;
    padding: 5px ;
  }
  .commenter {
    display: grid ;
    grid-template-columns: 1fr 1fr 1fr 1fr 1fr 5fr ;
    font-size: smaller ;
    font-style: italic ;
  }
  .content {
    background-color: rgb(100,100,100) ;
  }
  .label {
    color: hsla(160, 100%, 37%, 1);
  }
```

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.