📝Explanation of Microblog App (Vue 3 Composition API)

📦Overview

This is a small microblog app built in **Vue 3** using the **Composition API**. It shows horizontal cards for posts, lets users like posts, and add new posts — all in a single HTML file, without build tools.

🛠️File Structure

All code is in one file: `index.html`:

- CSS: for styling cards, buttons, etc.
- HTML: container `<div id="app"></div>` where Vue mounts.
- JS: Vue components, logic, and mounting.

🕔Styling (CSS)

```
body { ... }
```

- Set font, background color, and layout width.

```
.card-list {
  display: flex;
  flex-wrap: wrap;
  gap: 12px;
}
```

- Arrange cards horizontally and wrap to next line.

```
.card { ... }
```

- Styles each card: white background, shadow, rounded corners, hover effect.

```
.content span { color: #007BFF; }
```

- Styles hashtags inside content in blue.

```
.actions { text-align: right; }
```

- Aligns like button to the right inside each card.

```
button, input { ... }
```

- Style add button, like button, and input box.

🧰Vue Components (with Composition API)

✂️ `store` and `testPosts`

```
const store = Vue.reactive({ posts: [] });
const testPosts = [ ... ];
```

- `Vue.reactive()` makes an object reactive.
- `testPosts` provides initial data when app loads.

✒️ `Controls` Component

Purpose: input box + "Add" button to add a new post.

```
const Controls = {
  template: `<input v-model="newPost"> <button @click="add">Add</button>`,
  setup(props, { emit }) {
    const newPost = Vue.ref('');
    function add() {
      if (newPost.value.trim() !== '') {
        emit('new-post', { id: Date.now(), content: newPost.value, likes: 0 });
        newPost.value = '';
      }
    }
    return { newPost, add };
  }
};
```

- `setup()` is where logic lives in Composition API.
- `Vue.ref('')` → creates reactive text input.
- `emit('new-post', post)` sends new post to parent.

🛏️ `Hashtag` Component

Purpose: display each post as a card + like button.

```
const Hashtag = {
  props: ['post'],
  template: `...` ,
  setup(props) {
    const formattedContent = Vue.computed(() => {
      return props.post.content.replace(/(#\w+)/g, '<span>$1</span>');
    });
    function like() { props.post.likes += 1; }
    return { formattedContent, like };
  }
};
```

- Receives `post` as prop.
- `Vue.computed()` → highlights hashtags in text.
- `like()` increases likes count.

🌂 `App` Component (main)

Purpose: root component that shows all posts and controls.

```
const App = {
  template: `
    <h2>Microblog</h2>
    <controls @new-post="addPost"></controls>
    <div class="card-list">
      <hashtag v-for="post in posts" :key="post.id" :post="post"></hashtag>
    </div>` ,
  setup() {
    const posts = Vue.ref([...testPosts]);
    function addPost(post) {
      posts.value.push(post);
      store.posts.push(post);
    }
    return { posts, addPost };
  }
};
```

- `Vue.ref([...testPosts])` → reactive list of posts.
- `addPost()` adds new post to the list.
- Uses `<controls>` and `<hashtag>` components.

💣Mounting Vue App

```
const app = Vue.createApp(App);
app.component('controls', Controls);
app.component('hashtag', Hashtag);
app.mount('#app');
```

- Create Vue app from `App`.
- Register child components.
- Mount to `#app` div.

🔗Why Composition API?

- All logic (`ref`, `computed`, methods) is grouped inside `setup()`.
- Clear, modular, reusable, and TypeScript-friendly.

📦Result:

- Modern horizontal cards for posts.
- Like button on each card.
- Input to add new posts.
- Hashtags styled in blue.
- All reactive, thanks to Vue 3.

✨If you'd like, I can add a diagram or visual flow next!