📚Vue.js Composition API – Teaching Plan using Microblog Project

🔗Goal

Students will learn: - What Vue.js is and why it's reactive - Composition API (`setup()`, `ref`, `computed`) - Build a real microblog app: add posts, like posts, highlight hashtags - Use components, props, and events - Basic CSS styling and layout

---

🪜 Step by Step Lesson Plan

✂️Introduction & Setup

- Explain Vue, reactive UI, Composition API vs Options API.
- Create empty `index.html` with Vue CDN:

```
<!DOCTYPE html>
<html>
<head>
  <title>Vue Microblog</title>
  <script src="https://unpkg.com/vue@3/dist/vue.global.js"></script>
</head>
<body>
  <div id="app"></div>
</body>
</html>
```

✒️First Component (Hello Vue)

- Create `App` component:

```
const App = {
  template: `<h2>Hello Vue Microblog!</h2>`,
  setup() {
    return {};
  }
};
const app = Vue.createApp(App);
app.mount('#app');
```

- Explain `setup()` and mounting.

🛏️Reactive State: Show posts

- Define posts with `ref`:

```
const posts = Vue.ref([
  { id: 1, content: "First post! #hello", likes: 0 }
]);
```

- Render with `v-for`:

```
<div v-for="post in posts" :key="post.id">
  {{ post.content }} - 👍 {{ post.likes }}
</div>
```

- Explain reactivity.

🌂Like Button

- Add button:

```
<button @click="like(post)">Like</button>
```

- In `setup()`:

```
function like(post) { post.likes += 1; }
return { posts, like };
```

- Explain event binding and reactive updates.

🌡️ Add New Post (Controls component)

- Create `Controls`:

```
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 };
  }
};
```

- Parent listens:

```
<controls @new-post="addPost"></controls>
```

- In `App`:

```
function addPost(post) { posts.value.push(post); }
```

🎒Highlight Hashtags (Hashtag component)

- Create `Hashtag`:

```
const Hashtag = {
  props: ['post'],
  template: `<div v-html="formattedContent"></div><button @click="like">👍
{{ post.likes }}</button>`,
  setup(props) {
    const formattedContent = Vue.computed(() =>
      props.post.content.replace(/(#\w+)/g, '<span style="color:blue">$1</
span>')
    );
    function like() { props.post.likes += 1; }
    return { formattedContent, like };
  }
};
```

- Explain `computed` and `v-html`.

🎓Register & Use Components

- Register:

```
app.component('controls', Controls);
app.component('hashtag', Hashtag);
```

- Use in template:

```
<div class="card-list">
  <hashtag v-for="post in posts" :key="post.id" :post="post"></hashtag>
</div>
```

- Explain props and data flow.

🎩CSS Styling

- Add flexbox for horizontal cards, shadows, rounded corners.
- Show live how CSS changes appearance.

🔗Wrap-up

- Review: `ref`, `computed`, events, props, components.
- Show how Composition API keeps logic clean.

🗒️Extensions

- Filter by hashtag
- Delete post
- LocalStorage
- Total likes in header

If you'd like, I can export this as PDF or add diagrams!