

Übungsblatt 06: GUI Komponenten mit Vue.js

1. Theoriefragen

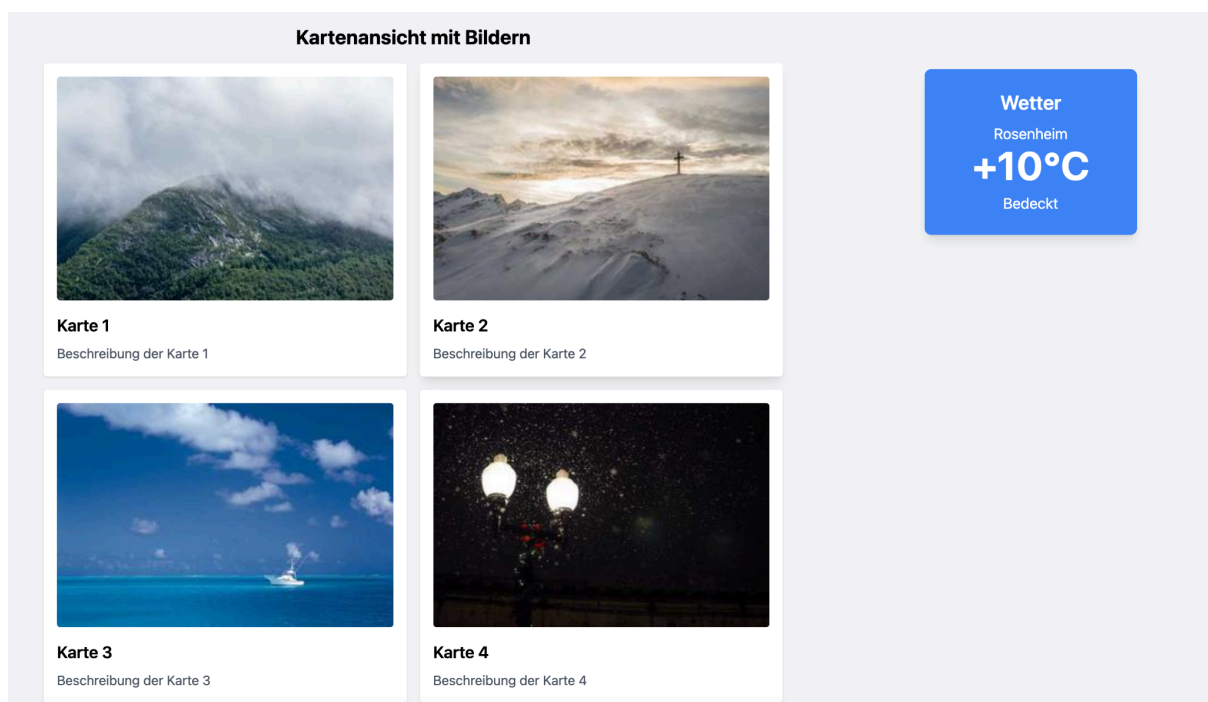
1. **Was ist eine GUI-Komponente?**
Beschreibe, welche Eigenschaften eine GUI-Komponente hat und warum sie in der Software-Entwicklung nützlich ist.
 2. **Was bedeutet Kontextfreiheit bei GUI-Komponenten?**
Erkläre, warum GUI-Komponenten kontextfrei sein sollten und wie sie dennoch mit anderen Komponenten kommunizieren können.
 3. **Wie funktioniert die Kommunikation zwischen Kind- und Eltern-Komponenten?**
Gehe darauf ein, welche Richtungen erlaubt sind und welche nicht? Erkläre wieso.
 4. **Was sind Single File Components (SFCs) in Vue.js?**
Erkläre, was eine Single File Component ist, welche Teile sie enthält und welche Vorteile sie haben.
 5. **Was sind Slots in Vue.js?**
Beschreibe die verschiedenen Arten und wann es sinnvoll ist, sie einzusetzen - und wann nicht.
-

2. Projekt Setup

Setzen Sie ein Projekt im Stil der vorhergehenden Übungen auf.
Für das Styling nutzen Sie weiterhin tailwind.css.

3. Praxisaufgabe

Erstellen Sie eine Vue.js Applikation und orientieren Sie sich dabei am vorgegebenen Design.
Nutzen Sie Tailwind CSS für das Styling.



Schritt 1: Statische Implementierung

- Setzen Sie Bilder und Wetterinformationen zunächst statisch.

Schritt 2: Dynamische Inhalte

- Nutzen Sie die folgenden APIs, um Inhalte dynamisch zu laden:
 - Bilder: <https://picsum.photos> (z. B. `https://picsum.photos/id/123/300/200`)
 - Wetter: <https://wttr.in> (z. B. `https://wttr.in/Rosenheim?format=%t+%C`)
-

Projektstruktur

1. App (App.vue)

Template:

- Orientieren Sie sich an den Vorlesungsfolien.

Script Setup:

- Laden Sie die benötigten Komponenten (`import X from 'y'`).
 - Definieren Sie reaktive Variablen mit `ref()`, z. B. `showDialog` und `selectedCard`.
 - Implementieren Sie eine Funktion, die das Öffnen des Dialogs ermöglicht.
-

2. Kartenliste (components/CardsList.vue) und Karte (components/Card.vue)

Templates:

- Orientieren Sie sich an den Vorlesungsfolien.

CardsList.vue

- Legen Sie eine reaktive Liste (`ref`) für die Karten an.
- Jedes Kartenobjekt enthält:
 - `image` (Bild-URL)
 - `title` (Titel)
 - `description` (Beschreibung)
- Implementieren Sie eine Funktion, die beim Klicken auf eine Karte das Event `'openDialog'` auslöst.

Card.vue

- Nutzen Sie `defineProps`, um die Attribute der Karte zu definieren (`card`).
-

3. Wetterleiste (components/WeatherBar.vue)

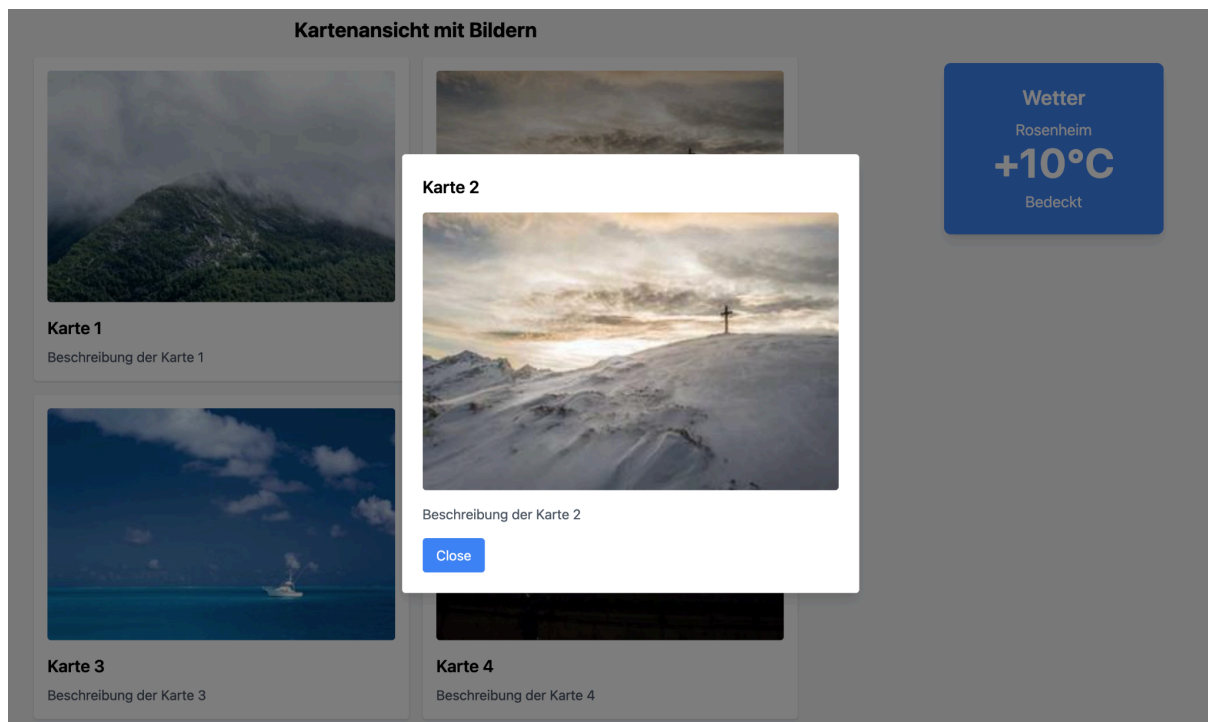
Template:

- Orientieren Sie sich an den Vorlesungsfolien.

Script Setup:

- Setzen Sie statische Werte für das Wetter.
 - Falls gewünscht, laden Sie Wetterdaten dynamisch in der `onMounted()`.
 - Implementieren Sie eine `fetchWeather`-Funktion, um die Wetterdaten dynamisch zu holen und in eine reaktive Variable zu speichern.
-

4. Dialog (components/Dialog.vue)



Template:

- Orientieren Sie sich an den Vorlesungsfolien.

Script Setup:

- Nutzen Sie `defineProps`, um eine `card` an den Dialog übergeben zu können.
- Implementieren Sie `defineEmits`, um das Event `'closeDialog'` auszulösen, wenn der Benutzer auf den Schließen-Button klickt.
- Integrieren Sie die Komponente in `App.vue`.
- Setzen Sie darin die notwendigen Werte:
 - Status, ob der Dialog geöffnet ist.

- Welche Karte ausgewählt wurde.
 - Verhalten beim Schließen des Dialogs (`closeDialog`).
-

Viel Spaß bei der Umsetzung!