Hochschule **Rosenheim**
University of Applied Sciences

Vorlesung GUI

Übungsblatt 02: Layouts

Theoriefragen

1. Erkläre die Ziele und Vorteile eines dynamischen Layouts.
2. Was sind Media Queries und wofür werden sie eingesetzt?
3. Vergleiche Flexbox und Grid Layout, beleuchte 3 Unterschiede.
4. Wie kann man komplexere Layouts umsetzen?

Praxisteil

Ziel:

Erstelle eine **Vue-Komponente**, die das Layout einer **einfachen Blog-Seite** umsetzt.

Struktur der Seite

Die Seite besteht aus **vier Hauptbereichen**:

1. **Ein Header**, der den Seitennamen und eine Navigationsleiste mit drei Links (Home, Articles, About) enthält. Die Links führen nirgends hin (`<a href="#" />`). Baue ein, dass die Links unterstrichen werden, wenn man darüber hovered.
2. **Ein Hauptinhalt**, in dem **Blog-Artikel** untereinander angeordnet sind.
3. **Eine Seitenleiste**, die zwei Sektionen enthält:
    - Eine **Kategorienliste** mit drei statischen Einträgen.
    - Eine **Über mich Sektion** mit einem kurzen Text.
4. **Ein Footer**, der eine einfache Copyright-Notiz enthält.

Anforderungen an das Layout

- Definieren eine **maximale Breite** für die Anzeige des kompletten Inhalts der Seite.

- Der **Header** soll oben bleiben und eine **horizontale Navigation** enthalten.
- Der **Hauptbereich** soll eine **Artikel-Ansicht** enthalten, die sich an die Bildschirmbreite anpasst.
- Die **Seitenleiste** soll **rechts neben dem Hauptbereich** sein.
- Der **Footer** soll **unter dem Hauptbereich** sein.

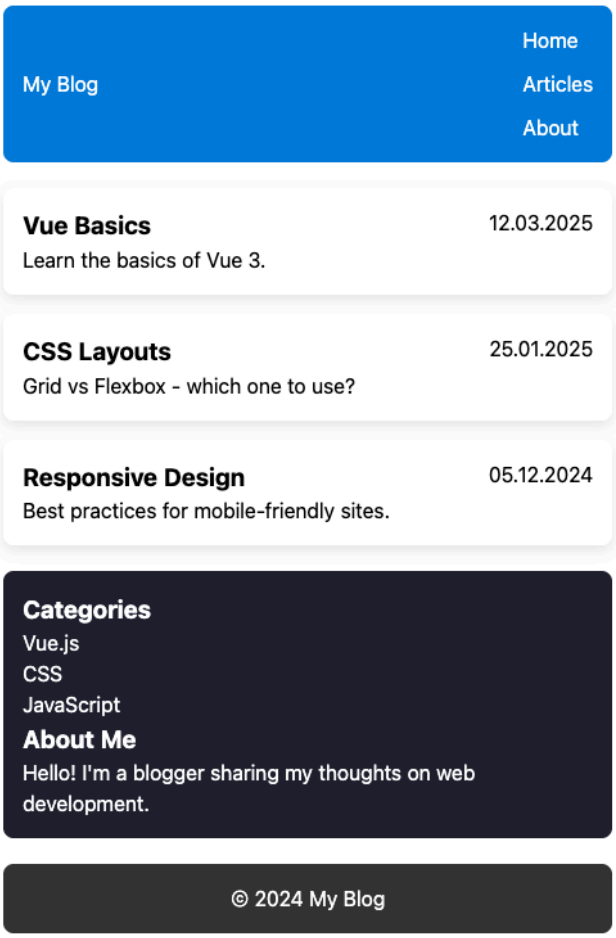
Anforderungen an die Umsetzung

- Das Layout soll mit `grid-template-areas` aufgebaut werden.
- Die Navigation soll ohne zusätzliche Funktionen als **einfache Liste** dargestellt werden.
- Du kannst **Beispiel-Blog-Einträge** sowie die **Kategorienliste** hard-coded in das Template einfügen. Wenn du schon Vue Erfahrung hast, gerne auch als Variablen.

Zusatzaufgabe: Responsive Design mit Media Queries

Erweitere die Komponente so, dass sie sich auf **mobilen Bildschirmgrößen** anpasst:

- Auf **kleinen Bildschirmen** soll die Sidebar **unter den Hauptinhalt** rutschen.
- Die **Navigation** soll auf mobilen Geräten **untereinander** angeordnet sein.



Zusatzaufgabe: Tailwind CSS

Baue deinen Code so um, dass du anstatt einfachem CSS die **Tailwind CSS** Klassenannotationen verwendest.

Hinweis: Mit Tailwind kannst du keine `grid-template-areas` verwenden.