

# Checkliste **Bestell App**

## (Lieferando Klon)

Bitte erfülle alle Punkte auf dieser Liste, bevor du das Projekt einreichst. Solltest du weitere Extras eingebaut haben, erwähne das kurz, damit sich die Mentoren dies bei Bedarf anschauen können.

### 1. Git-Workflow für dein Projekt

- ☐ Nutze GitHub von Anfang an. Denk dran: Dein GitHub-Profil ist deine Visitenkarte für Arbeitgeber – nutze diese Chance!
- ☐ Committe nach jeder Coding-Session
- ☐ Verwende klare, aussagekräftige Commit-Messages
- ☐ Verwende `.gitignore` verwenden, um unnötige Dateien auszuschließen
- ☐ Halte dein Repository aktuell und gepflegt

### 2. Funktionalitäten

- ☐ Es gibt einen Warenkorb/ Bild + Logo & Name von einem Restaurant / sowie eine Gerichteliste (grundsätzlich schickes design, angelehnt an Lieferando)
- ☐ Es gibt einen Header und einen Footer
- ☐ Es wird ein großes Bild z.B. vom Essen des Restaurants ganz oben angezeigt. Das Logo liegt halb über diesem Bild. (s. z.B. Lieferando)
- ☐ Restaurant Name, Bewertung, Beschreibung und Informationen zum Lieferpreis etc. sowie Sterne werden statisch angezeigt.
- ☐ Die auszuwählenden Gerichte (mindestens 5 Stück) werden dynamisch über JS aus einem Objekt gerendert.
- ☐ Es gibt einen Gerichteslider, der zu den jeweiligen Sektionen verlinkt, es sollte mindestens 3 verschiedene Kategorien geben mit jeweils mindestens 3 Gerichten, diese sollten von Bildern zur jeweiligen Kategorie getrennt sein. **(optional)**
- ☐ Man soll Gerichte in den Warenkorb hinzufügen können, indem man auf das Plus beim jeweiligen Gericht drückt.
- ☐ Im Warenkorb kann man nun für jedes einzelne enthaltene Gericht die Mengen erhöhen oder verringern sowie einzelne Gerichte komplett aus dem Warenkorb löschen.
- ☐ Im Warenkorb wird die Gesamtsumme angezeigt
  - ☐ ggf. mit Zwischensumme zwecks Lieferkosten etc. **(optional)**
- ☐ Man kann auf den bestellen Button drücken, danach sollte der Warenkorb geleert werden und eine Meldung kommen, dass man eine Testbestellung vorgenommen hat (kein Alert!).

- ☐ Wenn viele Gerichte im Warenkorb sind, sollte nichts aus dem Container herauslaufen.
- ☐ Der Desktop-Warenkorb sollte immer oben an der Seite angeheftet sein, außer man ist ganz oben oder unten auf der Seite. (sticky)

### 3. Code

- ☐ Aussagekräftige Namen für Funktionen und Variablen
- ☐ camelCase für die Benennung
- ☐ Code ist formatiert
- ☐ Höchstens 14 Zeilen pro Funktion
- ☐ Gleicher Abstand zwischen Funktionen (1 oder 2 Leerzeilen)
- ☐ Lagere HTML Templates aus in extra-Funktionen

### 4. Responsive

- ☐ Bis 320px Breite alles responsive ohne Scrollbalken
- ☐ Der Warenkorb ist jetzt rechts seitlich ausgeblendet, nun wird z.B. unten in der Mitte ein Button angezeigt, der den Responsive Warenkorb als Dialog öffnen kann und ggf. den Gesamtwert der Bestellung anzeigt.
- ☐ Im Responsive Warenkorb kann man wie im Desktop Warenkorb auch die Mengen der Gerichte erhöhen oder verringern sowie einzelne Gerichte komplett aus dem Warenkorb löschen und bestellen.

### 5. Sonstiges

- ☐ Favicon vorhanden
- ☐ Die Hauptseite muss **index.html** heißen, damit sie standardmäßig geladen wird.