

VAWi Web-Technologien, Sommersemester 2021

Studienleistung 2

Hinweise

- Die vollständige und korrekte Bearbeitung einer Aufgabe ergibt die volle Punktzahl dieser Aufgabe; in Ausnahmefällen werden halbe Punkte verteilt.
- Für diese Studienleistung erwarten wir einen **Bearbeitungsaufwand von insgesamt etwa 60 Stunden**, da wir von drei Personen pro Team ausgehen. Dies ist als Richtwert zu verstehen und keinesfalls als Unter- oder Obergrenze.
- In Verdachtsfällen behalten wir uns vor, Plagiate von der Benotung auszunehmen. Wir werden in einem solchen Fall Kontakt mit den Betroffenen aufnehmen.
- *Programmieraufgaben* sollen in nachvollziehbar kommentierter Form im Quelltext abgegeben werden. Verwenden Sie keine anderen als die angegebenen Bibliotheken, soweit sie nicht zum Standardsprachumfang gehören. Nicht lauffähige Programme können nicht bewertet werden. Fügen Sie gegebenenfalls eine README-Datei hinzu, die beschreibt, wie man Ihre Lösung kompilieren bzw. ausführen kann.

Abgabe der Lösungen

- *Abgabetermin* **05. Juli 2021, 23:55 Uhr**
- *Abgabemodus* Einreichung auf der Online-Lernplattform im entsprechenden Kurs; der Server ist erreichbar unter <http://lms.vawi.de/vawi/>.
- *Dateiformat der Abgabe* Die Abgabe erfolgt durch den Upload eines ZIP-Archivs, das alle notwendigen Dateien beinhaltet. Der Dateiname des ZIP-Archivs sollte das Namensschema **nachname1_nachname2_nachname3_02.zip** einhalten.
- *Update der Lösung* Bis zur oben angegebenen Deadline können Sie Ihre Lösung beliebig oft durch eine neue (korrigierte) Fassung ersetzen. Bitte beachten Sie, dass dabei die zuvor hochgeladene Lösung überschrieben wird. Wir haben keine Möglichkeit, alte Fassungen wiederherzustellen!

**Wir wünschen Ihnen viel Erfolg
bei der Bearbeitung der Studienleistung!**

Aufgabe: Erstellen eines One-Pagers zur Visualisierung von Daten (60 Punkte)

Die aktuelle Corona-Lage erlaubt ein wenig Hoffnung auf Besserung. Da das vor allem an den Impfungen liegt, wollen Sie zusammen mit einer Freundin eine Übersicht über den aktuellen Stand der Dinge erstellen. In ihrem Eifer hat Ihre Freundin bereits eine Datenschnittstelle im Internet recherchiert (<https://11n.de/t12/public>) und einen ausführlicheren Mockup der zukünftigen Seite erstellt (s. Abb. 1, 2, 3 und 4). Außerdem hat sie auch eine Anforderungsliste erstellt, welche Ihre Webseite erfüllen soll.

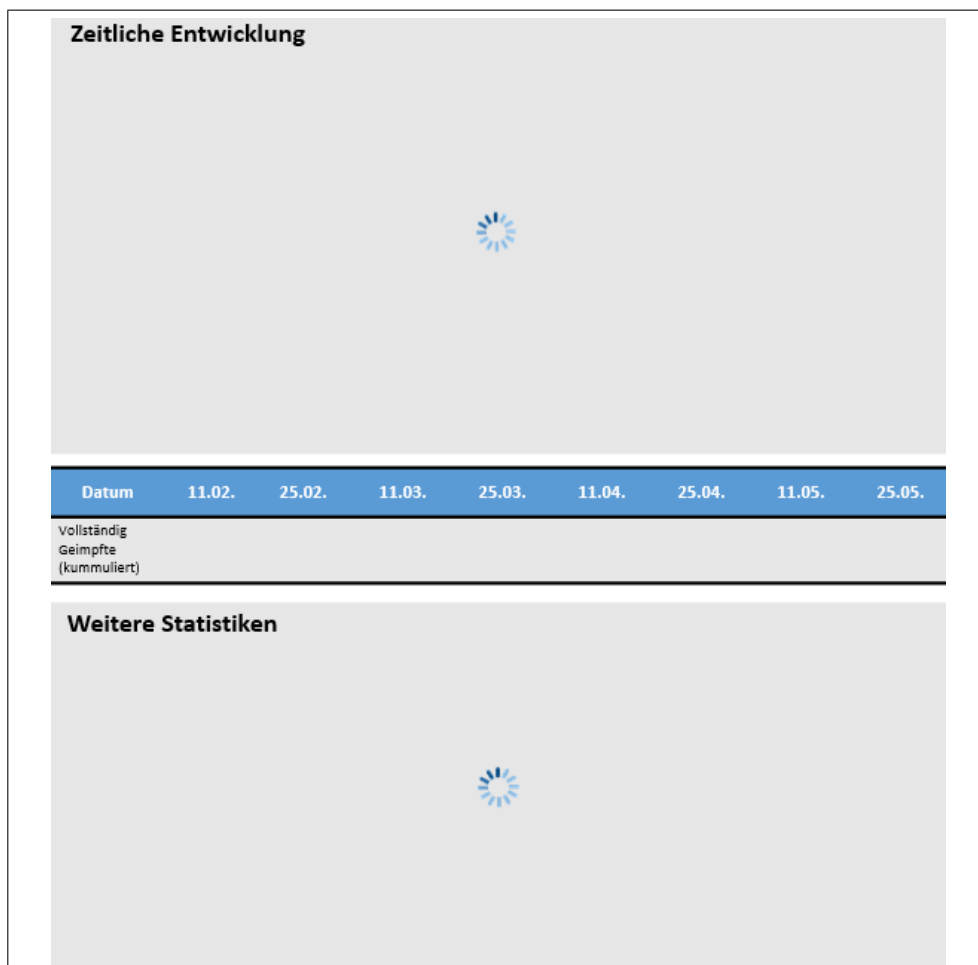


Abbildung 1: Während der simulierten Antwortzeit des Servers soll an Stelle der Diagramme das loader.gif angezeigt werden.

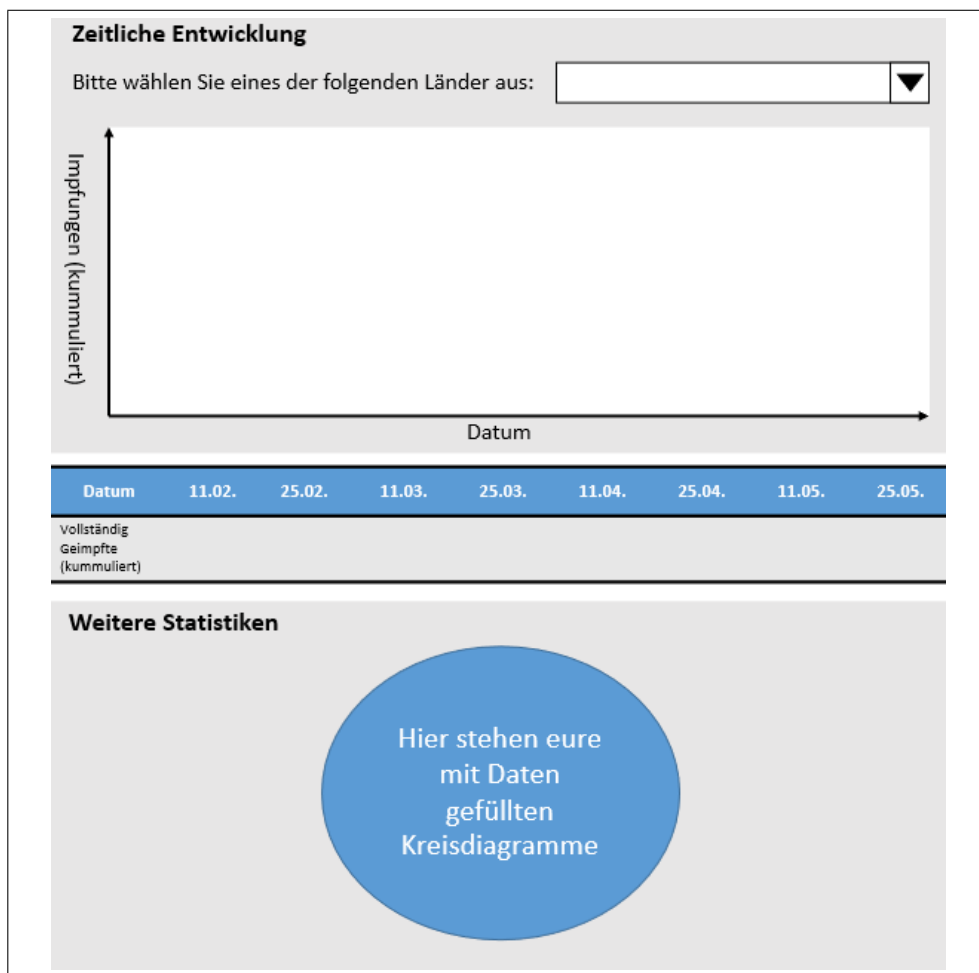


Abbildung 2: Ansicht der Visualisierungen nachdem die initialen Daten geladen wurden.

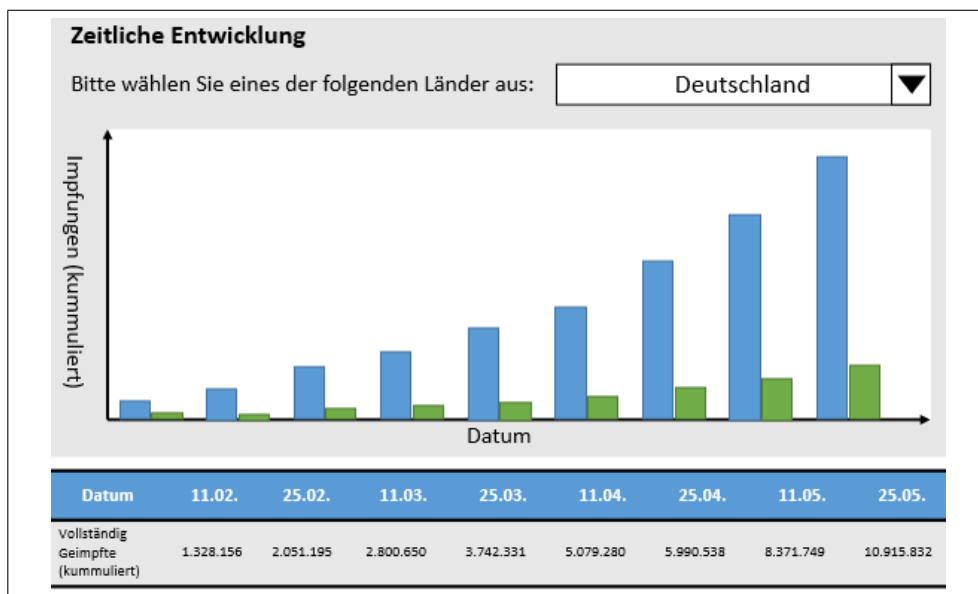


Abbildung 3: Ansicht des Balkendiagramms sobald ein Land ausgewählt wurde. Die Tabelle enthält in diesem Mockup noch keine Daten, in Ihrer Implementierung sollen aber die entsprechenden Zahlen enthalten sein.

CoronaVis – Visualisierung von Corona Trends

Allgemeine Informationen Zeitliche Entwicklung Weitere Statistiken

Auch im Jahr 2021 hat uns die Pandemie fest im Griff. Diese Seite bietet eine Übersicht über relevante Informationen.
[Mehr Informationen zum Corona Virus auf der Seite der Bundesregierung.](#)

Abbildung 4: Die Überschrift und der Navigationsbereich der Seite gefolgt von einem Bild inklusive des kurzen Informationstexts.

Entwickeln Sie unter Verwendung der vorgegebenen API eine Visualisierung mit der `d3.js`-Bibliothek. Insgesamt sollen drei verschiedene Visualisierungen und eine Tabelle realisiert werden. Zunächst soll die kumulierte Zahl der Personen, bei denen die

Impfserie begonnen wurde, und die Zahl der Personen, die vollständig geimpft wurden, eines ausgewählten Landes (Auswahl erfolgt über Dropdown-Feld) über den Zeitverlauf als Balkendiagramm dargestellt werden (y -Achse: kumulierte Zahl der Erstgeimpften/Vollständig Geimpften, x -Achse: Datum). Die kumulierten Zahlen der vollständig Geimpften über den Zeitverlauf sollen zusätzlich als Tabelle unterhalb des Balkendiagramms dargestellt werden.

Weiter unten auf der Seite sollen nebeneinander zwei verschiedene Kreisdiagramme dargestellt werden. Die Daten, die die Kreisdiagramme darstellen, können Sie sich selbst aussuchen, müssen aber ebenfalls aus der bereitgestellten API entnommen sein. Es sollen jedoch sinnvolle Daten sein, die bestenfalls kreative und/oder interessante Verhältnisse beinhalten.

API-Dokumentation

Die gegebene API, erreichbar unter <https://11n.de/t12/public>, hat folgende für die Bearbeitung der Teilleistung wichtige Endpunkte:

- <https://11n.de/t12/public/country/{country}/>: Gesamte über ein Land zur Verfügung stehenden Daten
- <https://11n.de/t12/public/country/{country}/vaccinations>: Historie der Impffzahlen seit Beginn der Aufzeichnung
- https://11n.de/t12/public/country/{country}/historical_data: Gesamte verfügbare Historie von Daten wie aktuelle Fallzahlen und Impffzahlen (**Achtung: vor allem frühe Daten haben teilweise keine Einträge zu Impfungen, hier */vaccinations* bevorzugen!**)
- <https://11n.de/t12/public/country/{country}/population>: Bevölkerungsanzahl
- https://11n.de/t12/public/country/{country}/population_density: Bevölkerungsdichte
- https://11n.de/t12/public/country/{country}/gdp_per_capita: BIP pro Kopf
- https://11n.de/t12/public/country/{country}/median_age: Median-Alter

Die für `{country}` einsetzbaren Werte sind die in der Datei `data/countries.json` zu findenden Kürzel.

Dateiübersicht

Für die Teilleistung wird die Datei `s12.zip` mit dem folgenden Inhalt im Virtuellen Campus bereitgestellt:

- `js`-Ordner: der noch leere Ordner, in dem Ihre JavaScript-Dateien platziert werden sollen.
- `css`-Ordner: der noch leere Ordner, in dem Ihre CSS-Dateien platziert werden sollen.

- **img**-Ordner: beinhaltet das Bild im GIF-Format, das angezeigt werden soll, bevor die Diagramme gezeichnet werden und damit die Antwortzeit des Servers simulieren.
- **data**-Ordner: beinhaltet die JSON-Datei, welche die Kürzel und ausgeschriebenen Namen der Länder enthält, die im Rahmen der Teilleistung visualisiert werden sollen.
- **index.html**-Datei: das Template für ihre Webseite, in dem sowohl Bootstrap als auch d3 bereits eingebunden sind.

Vorgehen und Anforderungen

- Verwenden Sie die vorgegebene **index.html**. Die Erstellung zusätzlicher HTML-Dateien ist nicht erlaubt. Zur Gestaltung soll ausschließlich Bootstrap und eigener CSS-Code verwendet werden, wobei auf eigenen CSS-Code so weit wie möglich verzichtet werden soll.
- Ihre Seite soll insgesamt in vier Bereiche unterteilt sein. Ganz oben soll sich eine Überschrift und die Navigationsleiste befinden. Darunter soll ein Bild und ein kurzer Informationstext zu den COVID-Impfungen sowie eine Verlinkung zur Corona-Seite der Bundesregierung platziert werden. Anschließend sollen die Bereiche für das Balkendiagramm inkl. Tabelle sowie für die beiden Kreisdiagramme folgen.
- Die Überschrift sowie die Navigationsleiste sollen fixiert sein. Außerdem soll über den Klick auf einen Menüpunkt an die entsprechende Stelle auf der Seite gesprungen werden.
- Das Anfordern der Daten soll mit asynchronen Requests per Fetch API umgesetzt werden. Berücksichtigen Sie auch den Fall, dass die angeforderte Datei bzw. die angefragte Route nicht zur Verfügung steht und geben Sie dann eine Fehlermeldung aus.
- Verändern Sie bis auf die **index.html**-Datei keine der vorgegebenen Dateien!
- Ihre Implementierung soll folgende Anforderungen erfüllen:
 - Beim Aufruf der HTML-Seite wird ohne Nutzerinteraktion der Request zum Holen der Daten für die Diagramme gestartet. Während auf die Antwort gewartet wird, soll an Stelle der Diagramme das animierte Bild `loader.gif` angezeigt werden (siehe Abb. 1).
 - Simulieren Sie unter Verwendung der `setTimeout()`-Funktion¹ eine Antwortzeit des Servers, welche zufällig bestimmt wird und zwischen einer und drei Sekunden liegt. In Abb. 1 sehen Sie, wie die Seite beim Warten aussehen könnte.
 - Sobald die Antwort des Servers verfügbar ist, soll der Nutzer über ein Dropdown-Feld eines der hinterlegten Länder auswählen können. Die Achsenbeschriftungen des Balkendiagramms können als feste Werte definiert werden.
 - Unter der Tabelle sollen die beiden Kreisdiagramme dargestellt werden.

¹https://www.w3schools.com/jsref/met_win_settimeout.asp, letzter Abruf: 7. Juni 2021

- Da es sich bei den Diagrammen um SVG-Vektorgrafiken handelt, soll die Größe des Diagramms nicht statisch vorgegeben sein, sondern sich dynamisch an die Bildschirmbreite anpassen. Außerdem soll Ihre Seite mobil nutzbar sein.

Webserver

Um lokale AJAX-Requests durchführen zu können und dabei keine Fehlermeldungen zu erhalten, benötigen Sie einen Webserver. Verwenden Sie als Webserver das Apache-Modul der neuesten Version von XAMPP. XAMPP ist für alle gängigen Betriebssysteme unter <https://www.apachefriends.org/de/index.html> verfügbar. Alternativ können Sie auch die Live-Server-Extension von Visual Studio Code nutzen. Geben Sie in Ihrer README-Datei an, wie Ihre Lösung ausgeführt werden kann.

Weitere Hinweise zur Studienleistung

- Nutzen Sie die Möglichkeiten der Konsolenansicht in den Entwicklungswerkzeugen von Firefox und Chrome zum Debuggen Ihres JavaScript-Codes.
- Erstellen Sie je eine HTML-, CSS- und JavaScript-Datei für Ihre Abgabe.
- Verwenden Sie neben d3.js keine weiteren JavaScript-Bibliotheken (auch kein jQuery!).
- Testen Sie das von Ihnen implementierte Event Handling auf möglich Fehler, die durch ungewöhnliche Nutzerinteraktionen entstehen könnten.
- Kapseln Sie zusammengehörige Funktionalität in Klassen, um Ihren Code besser zu strukturieren und die Wiederverwendbarkeit von Code-Ausschnitten zu erhöhen.
- Geben Sie in einer README-Datei an, ob Sie Ihre Umsetzung mit Firefox oder Google Chrome getestet haben. Dokumentieren Sie in dieser README-Datei bitte auch weitere Besonderheiten, z.B. wenn Ihre Umsetzung unvollständig ist oder sich unter bestimmten Umständen fehlerhaft verhält.
- Ihr Abgabe-ZIP-Archiv muss alle zur Ausführung notwendigen Dateien beinhalten, nicht nur solche, die Sie neu erstellt haben.