

## Prüfungsformalitäten

### Umfang des Portfolios

Die Portfolioprüfung beinhaltet die beiden Vorlesungen „Webprogrammierung“ und „Verteilte Systeme“ aus dem dritten und vierten Semester. Das Prüfungsportfolio setzt sich deshalb aus folgenden Bestandteilen zusammen, die wie folgt verrechnet werden:

Artefakt	Gewichtung
3. Semester: Webprogrammierung	
Lernkontrollen in Moodle	2
Dieses Aufgabenblatt	4
4. Semester: Verteilte Systeme	
Lernkontrollen in Moodle	1
Neues Aufgabenblatt	3
Übergreifend	
Einleitung	1
Reflexion	1

### Bearbeitung in Eigenleistung

Grundsätzlich sind alle Artefakte in Eigenleistung zu erstellen. Die Artefakte müssen den persönlichen Wissensstand widerspiegeln und soweit wie möglich einen individuellen Lösungsweg erkennen lassen. Zwar ist es durchaus erlaubt, sich in Form von Rückfragen oder Internetrecherche Hilfe zu holen. Die abgegebenen Lösungen müssen aber dennoch selbst ausgearbeitet worden sein, um gewertet werden zu können.

### Export der Lernkontrollen in Moodle

Da alle Prüfungsartefakte archiviert werden müssen, erhalten alle Studierenden nach Abschluss des Semesters ein PDF-Dokument mit den exportierten Lernkontrollen aus Moodle. Dieses Dokument muss dem Portfolio vor Abgabe hinzugefügt werden. Wir bitten daher, mit der Abgabe des Portfolios zu warten, bis das PDF-Dokument verschickt wurde.

### Einleitung und Reflexion

Einleitung und Reflexion müssen erst nach dem vierten Semester erstellt und abgegeben werden, da sie sich auf beide Semester beziehen. Sollten Sie das vierte Semester im Ausland verbringen und daher nicht an der Vorlesung „Verteilte Systeme“ teilnehmen können, geben Sie Einleitung und Reflexion bereits im dritten Semester ab. Hinweise zum erwarteten Inhalt beider Berichte finden Sie am Ende dieses Dokuments.

### Abgabe

Die Abgabe erfolgt bis zum Ende des Semesters bis zum Tag vor Beginn der nächsten Praxisphase.

# Aufgabenstellung

## Aufgabe 1: Grundlagen des Web

(10 + 10 + 5 Punkte)

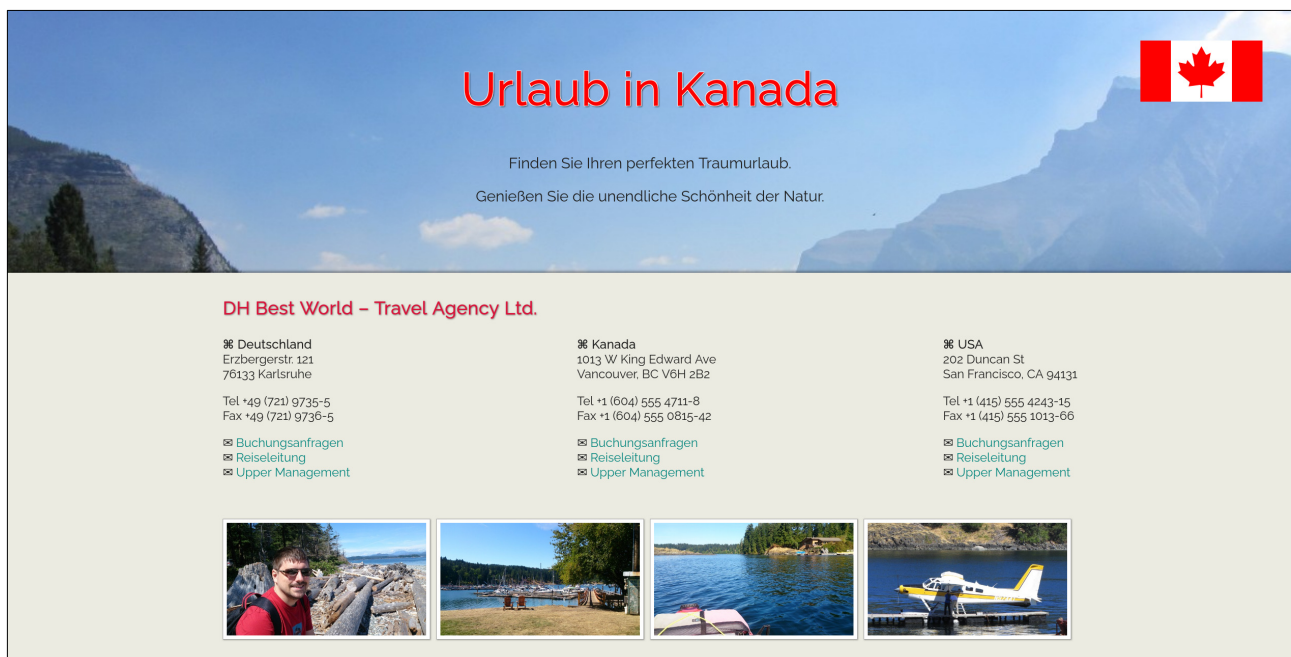
- a) Erklären Sie in eigenen Worten welche grundlegenden Technologien dem World Wide Web zugrunde liegen. Gehen Sie dabei von einer real existierenden Webseite oder Webanwendung aus und leiten Sie daraus je Technologie ein praktisches Beispiel ab, anhand dessen Sie die Funktionsweise und die Wechselwirkungen mit den anderen Technologien erklären. Die Antwort sollte mindestens zehn Sätze umfassen.
- b) Ordnen Sie dem folgenden Beispiel die typischen Bestandteile einer HTTP-Nachricht zu. Kennzeichnen Sie dabei alle optionalen Bestandteile und erklären Sie stichwortartig, in welchen Fällen Sie weggelassen werden dürfen.

Anfrage	Antwort
GET /favicon.png HTTP/1.1 Accept: */* Accept-Encoding: gzip, deflate Connection: keep-alive Host: www.wpvs.de User-Agent: HTTPie/1.0.3	HTTP/1.1 200 OK Accept-Ranges: bytes Access-Control-Allow-Origin: * Connection: Keep-Alive Content-Length: 15892 Content-Type: image/png Date: Fri, 18 Sep 2020 00:29:59 GMT ETag: "3e14-5aea78ea8390b" Last-Modified: Sun, 06 Sep 2020 16:25:59 GMT Proxy-Connection: keep-alive Server: Apache/2.4.38 (Debian)  +-----+   NOTE: binary data not shown in terminal   +-----+

- c) Setzen Sie eine gültige HTTP-Anfrage an einen beliebigen Webserver ab, ohne dabei auf einen Browser zurückzugreifen. Dokumentieren Sie die Anfrage und Antwort in einem Screenshot.

## Aufgabe 2: HTML und CSS

(10 + 10 + 10 Punkte)



Angenommen sie möchten eine Webseite für ein Reisebüro erstellen. Das Reisebüro möchte dabei, dass es sich um eine bildschirmfüllende Seite wie im obigen Screenshot handelt.

- a) Zählen Sie alle gestalterischen Einzelheiten auf, auf die Sie achten müssen, um den obigen Screenshot möglichst exakt nachzubauen.
- b) Wie würden Sie den HTML-Code der Seite strukturieren? Zeigen Sie das HTML-Gerüst der Seite auf und erklären Sie jeweils, warum Sie sich für den jeweiligen Code entschieden haben.
- c) Schreiben Sie ein zu Ihrem HTML-Code passendes Stylesheet, welches möglichst viele Merkmale des Screenshots realisiert. Beachten Sie dabei, dass nur die im Bild sichtbaren Merkmale umgesetzt werden sollen und die Seite zumindest auf großen Bildschirmen nicht scrollt.

### Aufgabe 3: Responsive Webdesign

(5 + 5 + 10 Punkte)

- a) Grenzen Sie die Begriffe „Responsive Webdesign“ und „Mobile First“ voneinander ab und erklären Sie anschließend, wie diese technisch realisiert werden können. Gehen Sie dabei insbesondere auch auf die benötigte Syntax ein. Ihre Antwort sollte mindestens zehn Sätze umfassen.
- b) Gegeben Sie der folgende Quellcodeausschnitt einer mit Bootstrap erstellten Webseite. Erstellen Sie jeweils eine Skizze, wie dieser auf großen Bildschirmen und auf Smartphones dargestellt wird:

```
<div class="container-fluid">
  <div class="row">
    <div class="col">
      <!-- Banner mit Hintergrundbild -->
    </div>
  </div>
  <div class="row">
    <div class="col-md">
      <!-- Link zu Unterseite 1 -->
    </div>
    <div class="col-md">
      <!-- Link zu Unterseite 2 -->
    </div>
    <div class="col-md">
      <!-- Link zu Unterseite 3 -->
    </div>
    <div class="col-md">
      <!-- Link zu Unterseite 4 -->
    </div>
  </div>
  <div class="row">
    <div class="col-md-7">
      <!-- Hauptinhalt 1 -->
    </div>
    <div class="col-md-5">
      <!-- Hauptinhalt 2 -->
    </div>
  </div>
  <div class="row">
    <div class="col">
      <!-- Fußbereich mit Impressum und so weiter -->
    </div>
  </div>
</div>
```

- c) Recherchieren Sie im Internet drei verschiedene CSS-Frameworks und stellen Sie diese kurz vor. Erklären Sie dabei anhand weniger Sätze, welche Probleme die Frameworks adressieren und welche Funktionen sie hierfür bereitstellen. Suchen Sie sich anschließend ein Framework aus und nehmen Sie eine begründete Einschätzung vor, in welchen Fällen Sie das Framework einsetzen oder nicht einsetzen würden.

### Aufgabe 4: JavaScript

(20 + 5 Punkte)

- a) Entwickeln Sie eine kleine Webanwendung entweder als clientseitige Single Page App oder als serverseitige Webanwendung mit node.js. Die Anwendung sollte aus mindestens zwei Seiten mit einheitli-

cher Gestaltung bestehen, zwischen denen beliebig hin und her gewechselt werden kann. Beispielsweise könnte die erste Seite eine Liste mit Datensätzen beinhalten, die auf der zweiten Seite im Detail angezeigt werden können. Es ist jedoch auch jede andere Funktionalität denkbar, so lange die Anwendung folgende Anforderungen erfüllt:

- Die Anwendung muss mit HTML, CSS und JavaScript entwickelt werden.
- Sämtliche Quellcodes sollen selbst entwickelt und nicht aus dem Internet kopiert werden.
- Der Quellcode sollte grundsätzlich objektorientiert strukturiert werden.
- Zusätzlich sollte der Quellcode ordentlich kommentiert und formatiert werden.
- Frameworks dürfen verwendet werden, plain old JavaScript ist aber auch okay.
- Die Seiten der App müssen ein einheitliches HTML-Gerüst und Layout besitzen.
- Die App sollte dabei ein liebevoll gestaltetes, ansprechendes Erscheinungsbild besitzen.
- Das Layout sollte gleichermaßen gut auf mobilen als auch stationären Endgeräten aussehen.
- Die Daten der Anwendung müssen nicht persistent (in einer Datenbank) gespeichert werden.

Funktional muss die Anwendung nicht komplex sein. Die umgesetzten Funktionen sollten aber durchgängig fehlerfrei sein. Vom Aufwand her sollte sie in etwa einer Anwendung zum Verwalten von „Lustigen Taschenbüchern“ entsprechen, bei der auf einer Seite eine Liste aller vorhandenen Datensätze und auf der anderen Seite ein Erfassungsformular zum Anlegen oder Bearbeiten enthalten sind. Aber auch kleine Spiele wie „Siebzehn und vier“ mit einer Seite für das Spiel und einer anderen für den Highscore sind in Ordnung.

b) Erklären Sie, warum Sie sich entweder für eine Single Page App oder eine serverseitige Webanwendung entschieden haben, indem Sie die Vor- und Nachteile beider Architekturen für Ihre Anwendung beschreiben. Ihre Antwort sollte zu beiden Architekturen möglichst drei Vorteile und drei Nachteile, die jeweils mit drei Sätzen beschrieben werden, nennen. Beschreiben Sie dabei auch grob, wie der Quellcode der Anwendung im Falle der jeweils anderen Architektur aufgebaut sein müsste.

# Einleitung und Reflexion

**Hinweis:** Die beiden Berichte sind erst im vierten Semester zusammen mit den Artefakten für „Verteilte Systeme“ abzugeben. Nur, wenn Sie die Vorlesung im vierten Semester zum Beispiel wegen Auslandsaufenthalt nicht besuchen können, geben Sie die Berichte schon im dritten Semester ab.

## Umfang und Inhalt der Einleitung

Die Einleitung kann besonders schlank ausfallen. Sie soll lediglich die zur Bewertung eingereichten Artefakte (Lernkontrollen, Aufgabenlösungen, usw.) aufzählen.

## Umfang und Inhalt der Reflexion

Lassen Sie die das komplette Modul (bestehend aus den Vorlesungen „Webprogrammierung“ und „Verteilte Systeme“, sofern Sie diese besucht haben) im Reflexionsbericht Revue passieren und zeichnen Sie dabei Ihren individuellen Lernweg nach. Verwenden Sie hierfür die in Moodle bereitgestellte Dokumentenvorlage mit den vorgegebenen Formatierungseinstellungen und achten Sie darauf, dass ein neuer Absatz nur durch einmal ENTER-Drücken erzeugt wird. Der Bericht sollte mindestens 1,5 Seiten umfassen und auf folgende Fragestellungen eingehen:

- 1) Mit welchen Voraussetzungen sind Sie in die Vorlesungen gestartet und welchen Lernweg haben sie im Verlauf der beiden Semester genommen?
- 2) Mit welchen Inhalten haben Sie sich warum sehr intensiv oder nur oberflächlich beschäftigt?
- 3) Inwiefern sind Ihnen verteilte Anwendungen schon früher im privaten und beruflichen Alltag begegnet und in wieweit verstehen Sie nun besser, wie diese funktionieren?
- 4) In welchem Zusammenhang werden Sie im Beruf voraussichtlich an der Konzeptionierung, Erstellung, Einführung, Dokumentation, Beratung, Schulung, Vertrieb oder Nutzung derartiger Anwendungen beteiligt sein?