|  |  |  |
| --- | --- | --- |
| **Kurs:** Web Technology | | **Code:** W5-WEB |
| **Studiengang** | Bachelor Wirtschaftsinformatik | |
| **Studiengruppe** | BWI-A20 | |
| **Dozent/in** | Walter Rothlin | |
| **Leistungsnachweis(e)** | Einzelarbeit mit Fachgespräch | |
| **Präsenz** | **Selbststudium** | **Workload** |
| 28 h | 52 h | 80 h |

|  |
| --- |
| **Lernergebnisse** |

Die Studierenden

* können eine einfache Web-Applikation mittels HTML-5, CSS und JavaScript (JS) designen und programmieren
* können ein REST-API verstehen und einen Request absetzen
* können einen REST Service via AJAX in einer Web-Applikation einbauen
* können den DOM via Java
* Kennen weitere WebApp Frameworks (Bootstrap, Angular, React, Vuejs, Node.js, Apache Cordova, Ionic, Electron)
* PWA (Progressive Web Apps), Responsive Design

|  |
| --- |
| **Lehrinhalte** |

1. http-Protokoll (request und response), Session, Stateless / Statefull
2. HTML-5 mit Form-Elementen / CSS
3. Request- / Response Web-Client entwickeln
4. Java-Script / Events and Event-Handlers / DOM
5. REST-Service (Open Weather, search, maps, QR-Code) API verstehen und Requests vom Browser her absetzen und Response analysieren.
6. AJAX verstehen und eine einfache Applikation bauen (JS ruft REST-Service auf)
7. Internet of Things (IoT) am Beispiel von Home Automatisation

|  |
| --- |
| **Lehr- und Lernmethoden** |

* Stoffvermittlung Frontalunterricht, Übungen lösen im Präsenzunterricht und Selbststudium

|  |
| --- |
| **Leistungsnachweise** |

Eine Applikation anhand von vorgegebenen Requirements nach den Clean-Code Regeln in Einzelarbeit entwickeln.

Geprüft wird der Code anhand der Funktionalen und Qualitäts-Kriterien. Anhand eines Fachgespräches wird überprüft, ob der Student den Code und die darunter liegenden theoretischen Grundlagen verstanden hat. (Gewicht: 100%)

|  |
| --- |
| **Lehrmittel** |

Empfohlen

* Moodle-Kurs mit Links zu Online-Tutorials

|  |  |
| --- | --- |
| **Thema Block 1: http-Protokoll** | |
| Datum | Do 22.09.2022 |
| Zeit | **17:30 – 20:45** |

Vorbereitung

* Aufbau einer URL mit Parametern

Inhalte

* http-roundtrip
* Unterschied http – https / Verschlüsselung
* Google-Chrome als Entwicklungs- und Analyse-Tool

Nachbereitung

* Analyse einer bekannten Web-Seite / Web-Applikation

|  |  |
| --- | --- |
| **Thema Block 2: HTML-5 (XHTML) / CSS** | |
| Datum | Do 06.10.2022 |
| Zeit | **17:30 – 20:45** |

Vorbereitung

* Self-HTML HTML und CSS

Inhalte

* Struktur und Tags von XHTML: HTML, BODY, A, TABLE, IMG, H, BR,….
* Publizieren auf öffentlichem Web-Server

Nachbereitung

* Mind. 3 Seiten fertigstellen und publizieren

|  |  |
| --- | --- |
| **Thema Block 3: HTML-Formulare** | |
| Datum | Di 11.10.2022 |
| Zeit | **13:15 – 16:30** |

Vorbereitung

* Standard-Formular Elemente

Inhalte

* Formular-Elemente in eine HTML-Seite einbauen
* Unterschied Post / Get
* Aktion auslösen / Eventhandler
* Hidden-Fields / Cookies
* Java-Script und DOM

Nachbereitung

* Anmelde-Seite implementieren

|  |  |
| --- | --- |
| **Thema Block 4: Formhandler (1. Leistungsnachweis)** | |
| Datum | Do 13.10.2022 |
| Zeit | **17:30 – 20:45** |

Vorbereitung

* Formhandler Beispiel analysieren und hidden fields verstehen

Inhalte

* Eigene Formhandler Applikation bauen und publizieren (1.Leistungsnachweis)

|  |  |
| --- | --- |
| **Thema Block 5: REST-Service nutzen** | |
| Datum | Do 27.10.2022 |
| Zeit | **17:30 – 20:45** |

Vorbereitung

* REST-Service und Response Typen (z.B. Open Weather API studieren und AppID lösen)

Inhalte

* Einfache AJAX Applikation

Nachbereitung

* Eigene AJAX Applikation fertigstellen und testen

|  |  |
| --- | --- |
| **Thema Block 6: Serverseitige Programmierung FLASK (2. Leistungsnachweis)** | |
| Datum | Do 24.11.2022 |
| Zeit | **17:30 – 20:45** |

Vorbereitung

* FLASK Web-Framework kennenlernen

Inhalte

* Eine FLASK Applikation nach Vorgaben implementieren (2.Leistungsnachweis)

|  |  |
| --- | --- |
| **Thema Block 7: IoT Anwendungen** | |
| Datum | **Do 15.12.2022** |
| Zeit | **17:30 – 20:45** |

Vorbereitung

* JSON Response Strukturen vom Open Weather analysieren
* Dictionaries und Listen Bearbeitung i Python

Inhalte

* Bestehende IoT (Service API) analysieren (z.B. Shelly für Home Automation)
* Eigener REST-Service mit JSON Response implementieren