

Praktische Aufgabenstellung Block 1: Node Backend

In dieser Aufgabe werden Sie einen einfachen Webserver in der Programmiersprache JavaScript bauen. Es handelt sich hierbei also um eine Anwendung, die auf Anfragen des HTTP-Protokolls reagieren soll. Die Aufgabe stellen Sie dann am Ende des Tages dem Dozenten zur Abnahme vor.

Der einfache Webserver soll wie folgt umgesetzt werden:

- Wenn die URL <http://localhost:2000> aufgerufen wird, soll eine HTML-Seite "index.html" an den Browser zurückgegeben werden.
- Denken Sie zu Beginn daran, das Verzeichnis für die Nutzung mit NPM vorzubereiten und installieren Sie das Webframework express. Mit der Erweiterung nodemon lässt sich ein Server starten, der stetig auf Änderungen in den *.js-Dateien lauscht und dann nach einer Speicherung neu startet. So ersparen Sie sich das manuelle Neustarten des Webserver.
- Die Datei "index.html" enthält Links auf mindestens drei weitere HTML-Dokumente, die Sie bitte auch erstellen (zunächst leer). Es handelt sich also um eine sogenannte Multi-Page-Seite mit separaten HTML-Dateien.
- Verwenden Sie zur Gestaltung der Webseiten ein CSS-Framework wie
 - o Bootstrap (siehe <https://getbootstrap.com/>),
 - o Tailwind CSS (siehe <https://tailwindcss.com/>) oder
 - o Materialize (siehe <https://materializecss.com>).

Finden Sie heraus, wie sich das Framework in ihre Webseiten integrieren lässt. Nutzen Sie dann die vom Framework bereitgestellten HTML+CSS-Komponenten, um eine einheitliche und professionelle Darstellung zu erreichen.

- Erweitern Sie eine ihrer HTML-Seiten um eine dynamische Funktionalität zum Speichern eines Kommentars, d.h. der Nutzer der Seite kann in ein Textfeld einen Kommentar eingeben und in ein anderes Textfeld seinen Namen. Nachdem diese Daten an den Server gesandt wurden, werden die Daten dort in einem Array gespeichert. Die Serverdatei benötigt dafür einen POST-Endpunkt, der die Daten entgegennimmt.
- Auf einer weiteren HTML-Seite sollen alle im Array gespeicherten Kommentare und Namen angezeigt werden.
- Verwenden Sie für diese Aufgabe keine weiteren Web-Frameworks wie Vue, Angular oder React. Ziel ist es hier, ein grundlegendes Verständnis für Webserver in Node.js sowie serverseitiges Rendering zu erhalten.
- Bauen Sie Ihre Seite responsiv, d.h. erstellen Sie verschiedene Versionen für verschiedene Bildschirmbreiten (z.B. Handy und Desktop-PC).
- Überlegen Sie sich eine inhaltliche Idee für Ihre Webseite und bauen Sie ihre Webseiten entsprechend mit Bildern, Farben, Animationen aus.
- Welche Quellen, Tutorials oder auch generative KI wie ChatGPT verwenden Sie zum Erstellen dieser Aufgabe? Notieren Sie sich ihre Vorgehensweise, um sie später im persönlichen Gespräch mit dem Dozenten reflektieren zu können.

- Bewertungskriterien für diese Aufgabe sind:
 - Technisch saubere Umsetzung gemäß den o.g. Anforderungen und Funktionsfähigkeit
 - Klar strukturiertes, sauberes User Interface
 - Mündlicher Kompetenzeindruck aller Gruppenmitglieder zum gesamten Code: Stellen Sie sicher, dass Sie jede Zeile Ihres Codes vollständig verstanden haben! Dies ist in Zeiten von generativer KI von besonderer Bedeutung.
 - Gute Lesbarkeit und Sauberkeit der Codebasis
 - ggf. Kreativität oder Innovation in der technischen, gestalterischen Umsetzung
- Die Abgabe der Aufgabe erfolgt am Ende des Tages per Email an den Dozenten mit Link auf das eigene Github-Repository (public) oder als Zip-Datei mit allen Dateien exklusive dem Node-Modules-Ordner via <https://fex.hs-weingarten.de/>