

Projekt Last Minute Camping – Tent Edition

Zielsetzung

Für die Umsetzung unseres Projekts haben wir uns erneut an folgendem Modell orientiert:



Ziel war eine Datenbank mit der Programmiersprache „Python“ zu erstellen sowie diese ebenfalls via Python zu verwalten. Anschließend sollten die eingepflegten Daten veranschaulicht und mehr oder minder sinnvoll verwendet werden.

Planung und Entscheidungsfindung

Im ersten Schritt der Planung befassten wir uns mit der Frage, was wir darstellen wollen.\

Da im ersten Halbjahr bereits ein Datenbank-Projekt vorausgegangen war in welchem wir eine mySQL-Datenbank zum Thema „Last-Minute-Camping“ erstellten, war schnell klar, dass wir hier auch wieder ansetzen könnten.

Die Entscheidung war also gefallen.

Da wir in den Wochen zuvor bereits daran gearbeitet hatten eine Datenbank in Python zu erstellen war auch der Ausgangspunkt für das Projekt schnell gefunden.

Um das Projekt überschaubarer und anschaulicher zu halten, als auch aufgrund des knappen Zeitrahmens, entschieden wir uns, keine Kopie der Datenbank aus dem ersten Halbjahr zu erstellen, sondern eine neue, kompaktere Version dieser zu entwickeln. Die „Tent Editon“ war geboren, welche sich ausschließlich auf Camper mit Zelt fokussieren sollte.

Als Nächstes beschäftigten wir uns mit der sinnvollen Verwendung unserer Datenbank und der enthaltenen Daten. In unserem Projekt im ersten Halbjahr wurden viele Informationen in eine Datenbank eingepflegt und konnten auch jederzeit vom betreffenden Administrator verändert werden, aber von außen einsehen, geschweige denn als möglicher Interessent hiermit zu interagieren war leider nicht möglich. Aber wie wollten wir unsere Daten weiterverarbeiten? Da es sich um eine Übersicht zu Campingplätzen sowie deren freie Kapazitäten handelt war der logische Entschluss unsere Daten für interessierte Camper und Camperinnen zur Verfügung zu stellen. Als eine der geläufigsten Varianten wählten wir eine browserbasierte Webansicht, welche dem zukünftigen Kunden eine intuitive und übersichtliche Darstellung unserer Daten vermitteln würde. Die Umsetzung sollte in „VUE“ erfolgen. Dabei handelt es sich um ein sogenanntes Javascript-Framework, welches die Verknüpfung von HTML-, CSS- und Javascript-Elementen stark vereinfacht und so auch den Entwicklungsprozess beschleunigt. Der große Vorteil der von uns gewählten Darstellungsform ist zudem, dass unsere Daten in jedem gängigen Internetbrowser visualisiert werden können. Abschließend stellte sich die Frage, wie wir ein clientseitiges Frontend, welches durch den Webbrowser dargestellt wird mit unserer

Datenbank verbinden könnten. Im Normalfall regelt ein Backend den Austausch der Daten zwischen Frontend und der Datenbank. Dies geschieht z.B., um für Sicherheit zu sorgen, da der Quellcode hierdurch nicht einsehbar und manipulierbar wäre oder aber auch um Datenintegrität zu gewährleisten wobei das Backend jeden Datenaustausch validieren würde und dadurch verhindert, dass es bei zeitgleichen Zugriffen verschiedener Benutzer zu Dateninkonsistenzen kommen kann. Da die zeitlichen Fristen für das Projekt schon sehr knapp erschienen, verwarfen wir aber schnell den Plan ein echtes Backend zu integrieren und suchten nach einer Behelfslösung. Die Wahl fiel auf eine .JSON-Datei (JavaScript Object Notation) als Zwischenspeicher. Mit dem Entschluss war die grobe Planung abgeschlossen und die Umsetzung konnte beginnen.

Umsetzung

Wie in der Planung bereits erwähnt, war unser Ausgangspunkt eine im Voraus erstellte „Personendatenbank“ in welcher wir eine mySQL-Datenbank mit Python erstellten und verwalten. Diese Verknüpfung wurde durch die Python-Bibliothek „mysql-connector“ realisiert. Um die Administration der Datenbank in Python zu ermöglichen wurden folgende Funktionalitäten implementiert:

- Datensatz hinzufügen
- Datensatz in der Datenbank suchen