

Web Application Development

AdViz, Live Server, npm Projekt

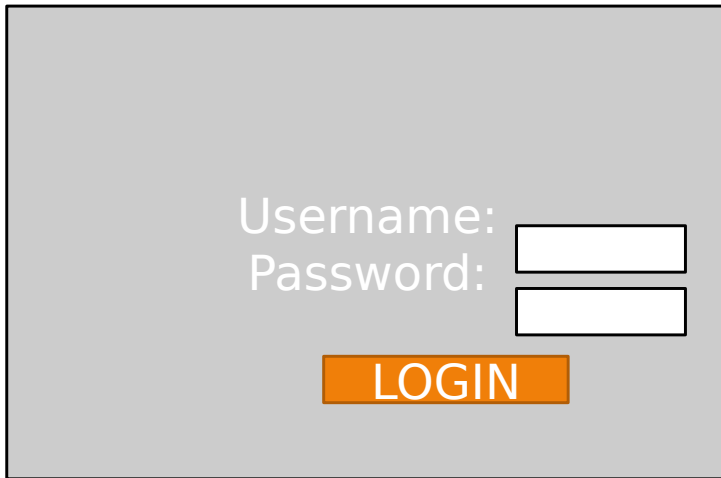
Martina Freundorfer

Semesterplan

- Ziel ist das Erstellen einer Webapplikation mit DB, Node-Backend und Frontend (html, css, JavaScript)
- “AddressViz” oder “AdViz” soll die Webapplikation heißen – wird Daten eines Adressbuchs auf einer Karte visualisieren
- Haben Kontaktdaten von Freunden, Familie, Firmen, Ärzten, etc. in einer Datenbank:
Name, Vorname, Strasse, PLZ, Stadt, Land, Longitude, Latitude, privat
- Wollen:
 - eine SPA (= Single Page Application), die im Frontend diese Daten auf einer Landkarte darstellen
 - 2 Nutzer-Typen soll die Anwendung kennen: Admin (Sie) und “Mutti”/“Vati”
 - Admin-Nutzer können neue Kontakt hinzufügen, alte löschen oder aktualisieren
 - Mutti/Vati -Nutzer dürfen nur “gucken”

Semesterplan

Login screen:

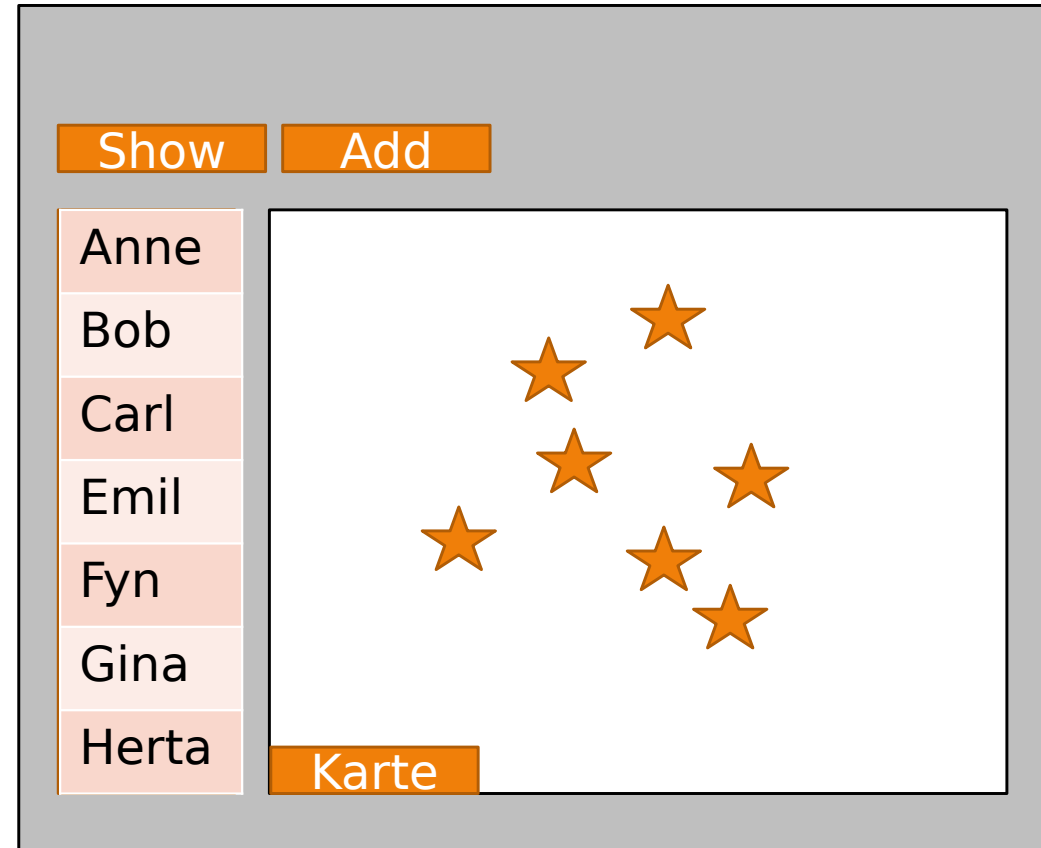


A login screen with a light gray background. It features two input fields: one for 'Username:' and one for 'Password:'. Below the password field is an orange button with the text 'LOGIN' in white capital letters.

Nutzerrollen:

1. Admin: darf Kontakte hinzufügen, ändern, löschen
2. Mutti/Vati: darf sich Kontakte nur anzeigen lassen und auch nur die nicht-privaten

Main screen für den Admin und Mutti/Vati

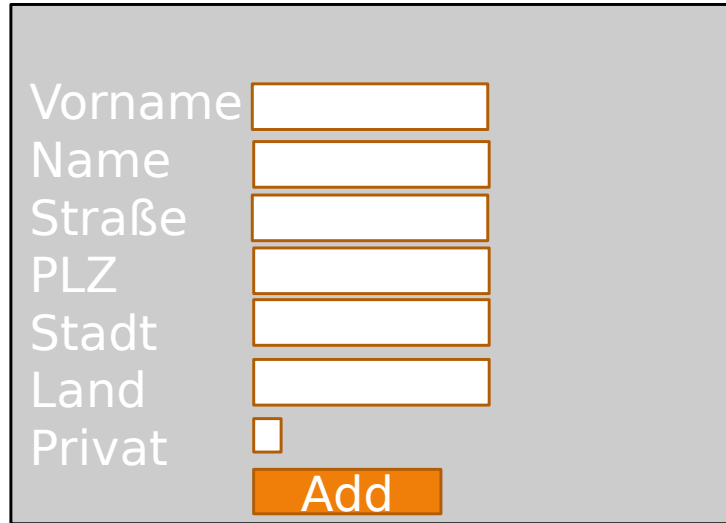


A main screen with a light gray background. At the top, there are two orange buttons: 'Show' and 'Add'. Below these is a list of names in a light pink box: Anne, Bob, Carl, Emil, Fyn, Gina, and Herta. To the right of the list is a large white area containing several orange stars. At the bottom right of this area is an orange button labeled 'Karte'.

Mutti/Vati sieht den "Add"- Button nicht, Kontakte gehören alle dem "admin", Mutti/Vati sieht nur die nicht-privaten.

Semesterplan

- Klicken auf “Add” öffnet Formular zum Ausfüllen:



Vorname

Name

Straße

PLZ

Stadt

Land

Privat ☐

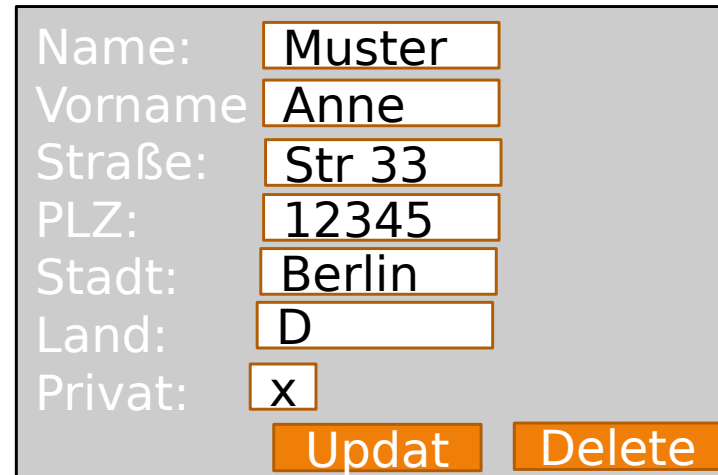
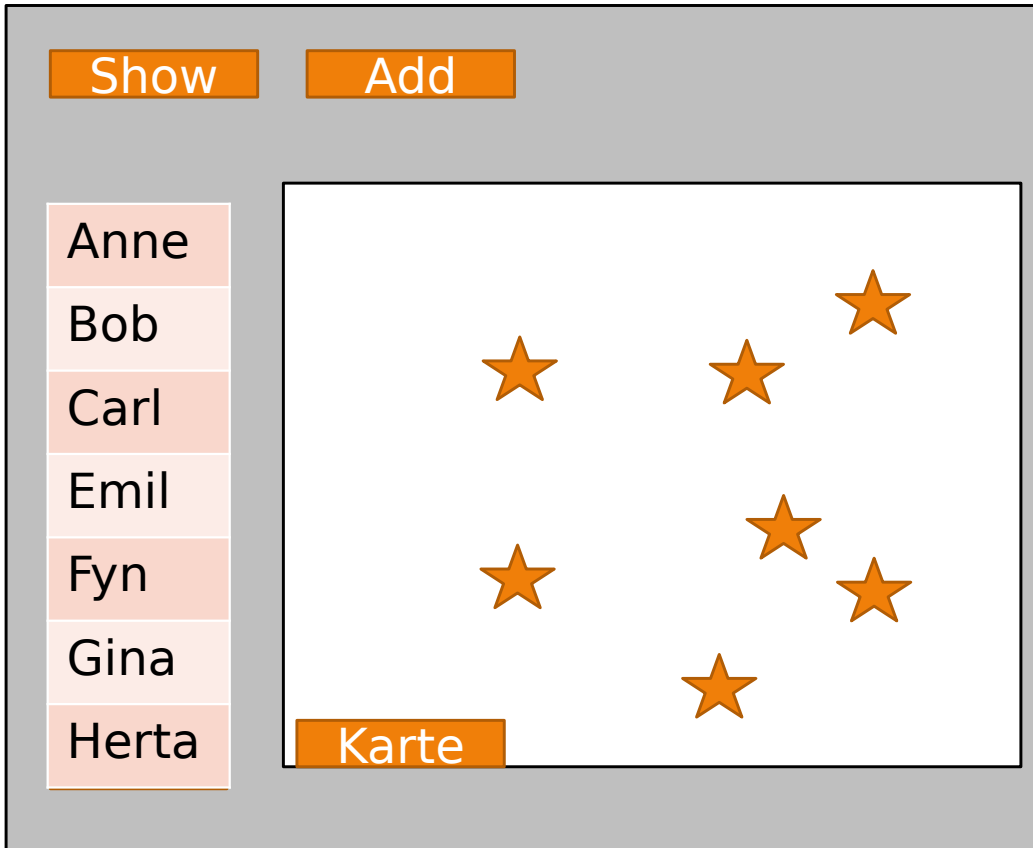
- Nutzer gibt Daten ein, klickt auf “Add” → Speichern in DB, neuer Kontakt wird auf Karte dargestellt
- Um eine Location auf einer Karte darzustellen, brauchen wir deren Lon/Lat.
- Wo kommen die Longitude/Latitude – Daten her?
- Antwort: Frontend (oder Backend) muss einen Service kontaktieren, der für eine gegebene Adresse die entspr. Long/Lat liefert (in Vorlesung über JavaScript)

Semesterplan

“Update” und “Delete” von Einträgen:

Als Admin und als Mutti/Vati kann ich auf einen Namen bzw. auf eine Markierung in der Karte klicken.

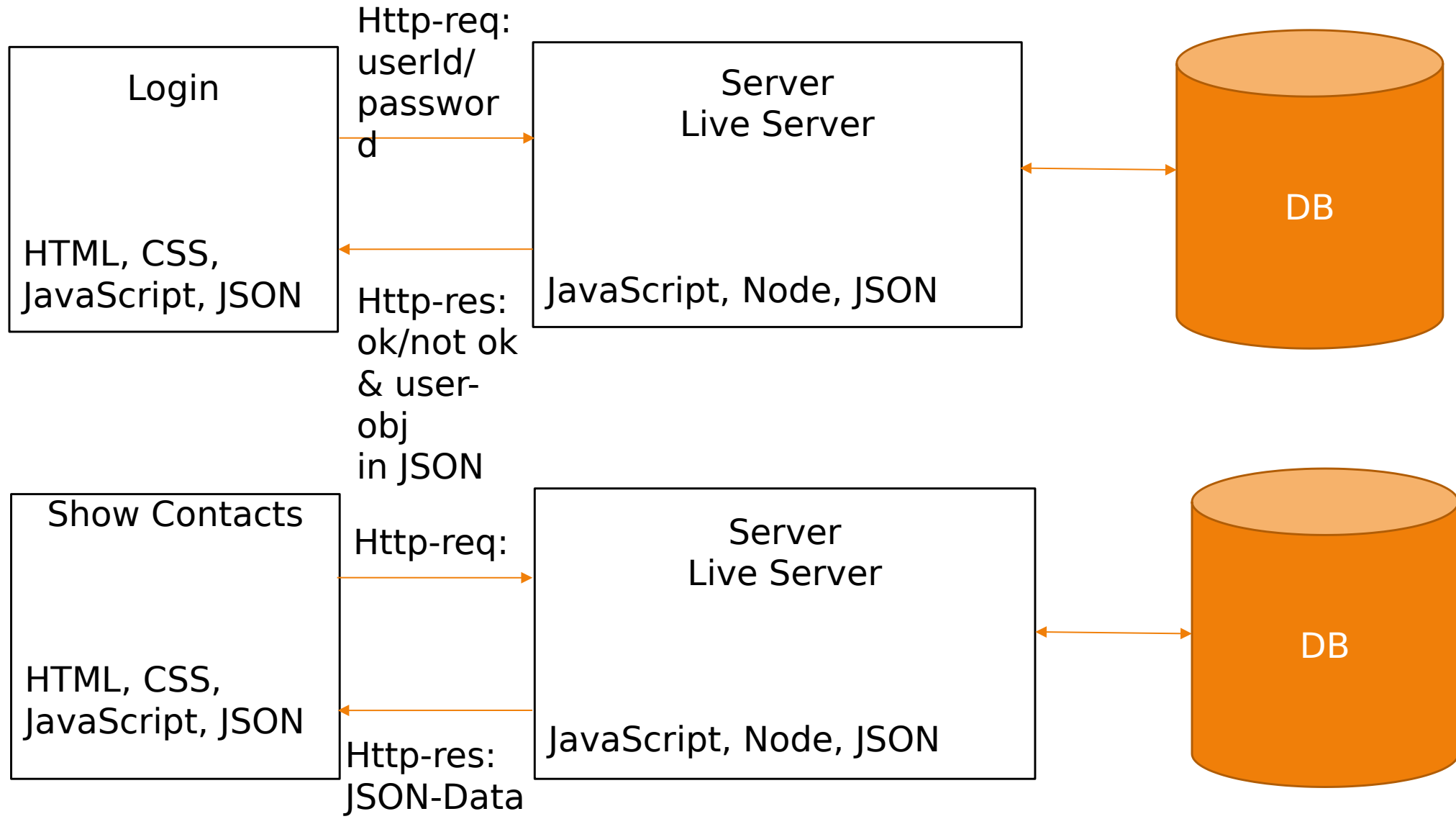
Dann öffnet sich diese Ansicht:



Mutti/Vati bekommt die “Update”/“Delete”- Buttons nicht. Mutti/Vati sieht nicht die privaten

Kontakte des Admins nicht.

Semesterplan



Live Server (CLI)

- Ist ein kleiner Webserver mit Live Reloading, d.h. einem Watcher, der mitgekommt, ob wir HTML/CSS/JS geändert haben
- Live Server mit npm installieren:

`npm install -g live-server`

Homepage <https://www.npmjs.com/package/live-server>

Alternative Live-Server Plugin (VSCode)

- Ist ein kleiner Webserver mit Live Reloading, d.h. einem Watcher, der mitgekommt, ob wir HTML/CSS/JS geändert haben
- Live Server mit ext installieren: `ext install ritwickdey.LiveServer`

Homepage <https://marketplace.visualstudio.com/items?itemName=ritwickdey.LiveServer>

Die erste Webapplikation

- Eine erste Webapplikation anlegen und bauen
- Lokal einen neuen Ordner namens “adViz” anlegen
- Im Terminal in diesem Ordner Npm init
- Git Verlinken

Legt package.json an (Node Package Manager = Paketverwaltung)

- Mkdir src (hier wird gearbeitet)
- Touch index.html

Unter-Verzeichnisse: css → touch style.css, img, js → touch script.js anlegen

- index.html editieren

Die erste Webapplikation

- Den Live Server im Projektverzeichnis starten
- `project> live-server`

`http://127.0.0.1:40721/src/` lädt automatisch die `index.html`

Alternativ Live-Server im VSCode

- `Project> code .`
- Startet VSCode im aktuellen Verzeichnis
- Rechtsklick auf die `index.html`

`http://127.0.0.1:5500/src/index.html` startet automatisch den Browser (inklusive Hot Deployment)

Exemplarische Projektstruktur

