

Prototyping Projektdokumentation

Name: Claudia Carvalho Paula

E-Mail: carvacla@students.zhaw.ch

URL der deployten Anwendung: <https://rent-a-rentner-deploy.vercel.app/>

1. Einleitung

Die Idee dieser Applikation ist eine Webanwendung zu bauen, welche es RentnerInnen ermöglicht ihre Erfahrung und Fähigkeiten gegen Bezahlung oder auch ehrenamtlich anbieten zu können. Die Plattform soll als Rentner benutzt werden können, um die Fähigkeiten anzubieten aber auch als Kunde, um die Fähigkeiten und Erfahrungen in Anspruch zu nehmen. So können Kunden anhand von Services, passende RentnerInnen finden.

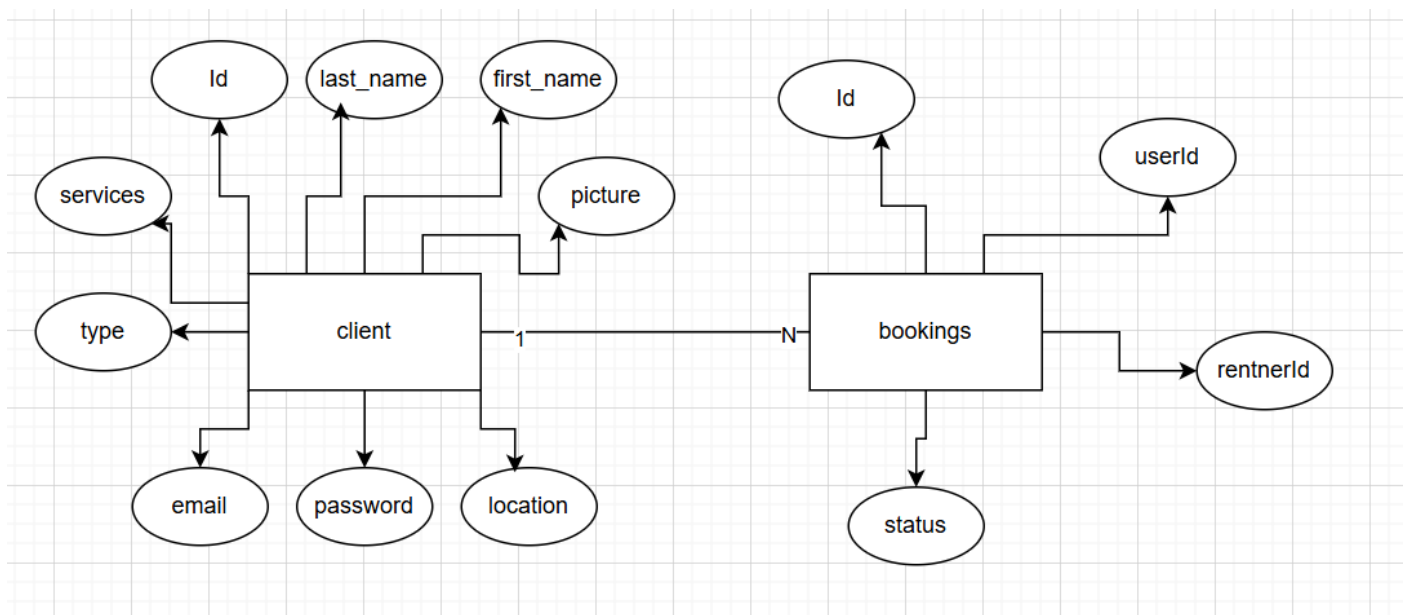
Die Grundfunktionen der Webanwendung sind:

- Benutzerregistrierung und Login: Ein Nutzer kann sich als Kunde oder als Rentner registrieren.
- Profilverwaltung: Auf einem Profil kann man das Profil wieder löschen.
- Rentner suche: Es gibt eine Übersicht aller Rentner, die sich auf der Plattform registriert haben.
- Detailansicht Rentner: Es gibt eine Detailansicht, bei der die einzelne Rentner mit detaillierten Informationen und Bild aufgerufen werden kann.
- Navigation: Je nach Login-Status sehen die Nutzer der Plattform unterschiedliche Navigationsoptionen.
- Profil Ansicht mit Löschfunktion: Nutzer können ihr Profil anschauen und das Profil wieder löschen.
- Logout: Nutzer können sich nach einem Login wieder ausloggen.
- Error Handling: Wurde vernachlässigt, da es sich um ein Prototype handelt.

Getestet kann das Ganze auch mit dem Login werden:

Email: test@test und PW: test Datenmodell

ERD von den Collections



Ein user kann viele bookings haben (Unterscheidung, ob clients oder rentner spielt in dem Fall keine Rolle)

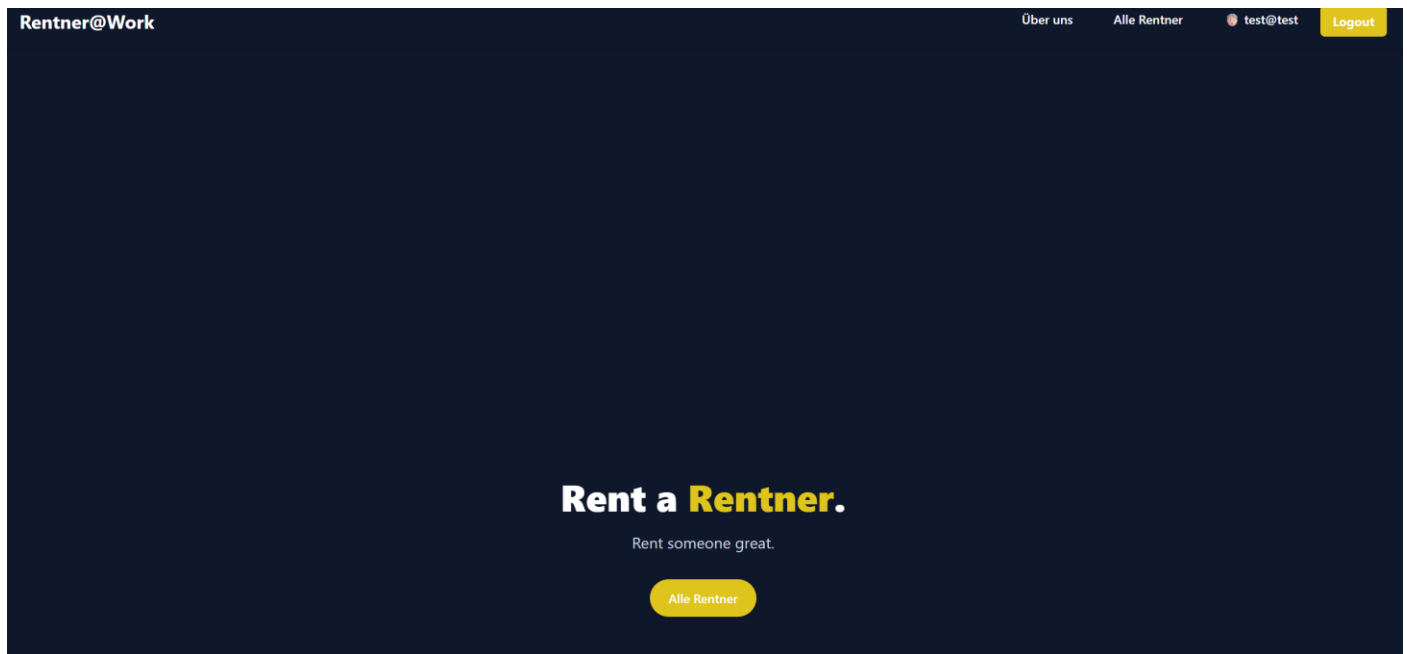
Ein Booking gehört genau zu einem client und einem rentner

2. Beschreibung der Anwendung

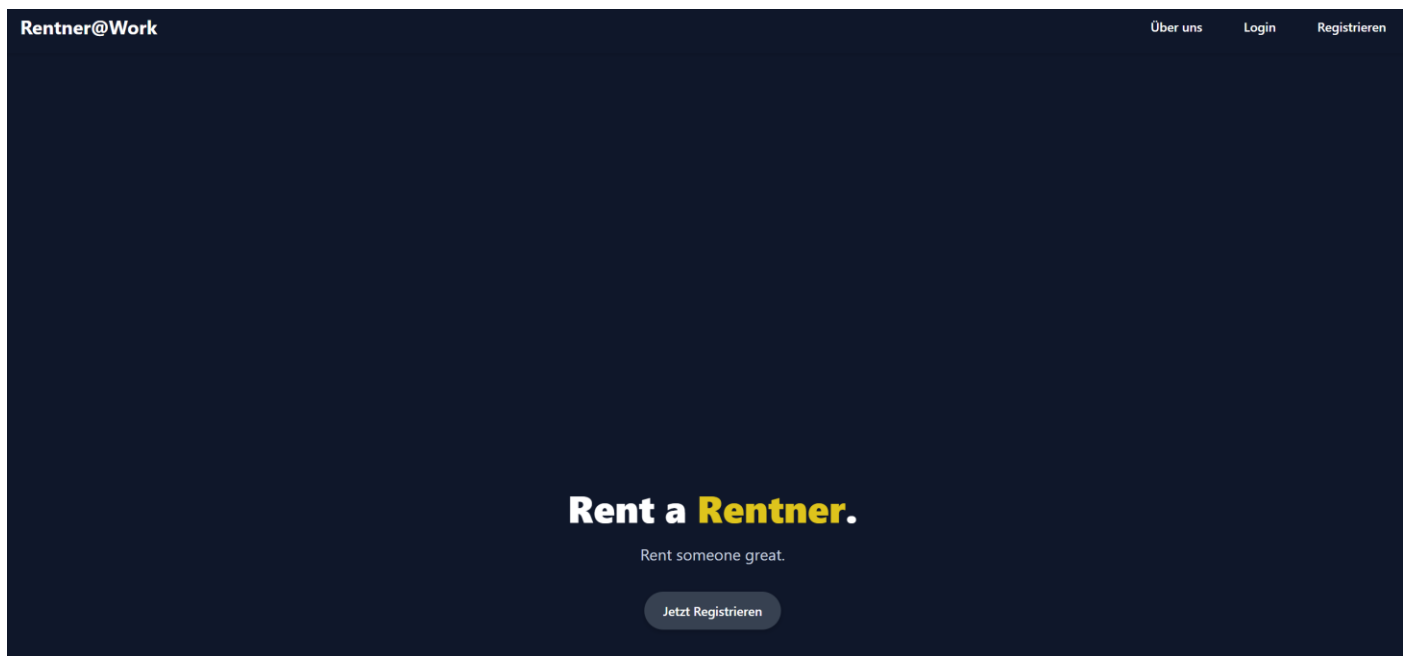
2.1. Startseite

Route: /

Wenn eingeloggt:



Wenn nicht eingeloggt:



Auf dieser Seite werden die Nutzer mit einem kurzen Text «begrüßt» und zur Registrierung oder Login aufgefordert. Über den Button «Alle Rentner» gelangt man zur Seite /rentner

Dateien:

- ./styles.css
- routes/+page.svelte

2.2. Über uns

Route: /about



Auf dieser Seite erhalten die Nutzer eine kurze Übersicht über die Plattform, mittels einer «Über uns» Seite.

Dateien:

- ../styles.css
- routes/about/+page.svelte

2.3. Gebucht

Route: /gebucht



Diese Seite dient zur Bestätigung einer Buchung. Des Weiteren wird diese im heutigen Zustand nicht gebraucht.

Dateien:

- ../styles.css
- routes/gebucht/+page.svelte

2.4. Login

Route: /login

Rentner@Work

Über uns Alle Rentner Login Registrieren

Ihr Login.

E-Mail

Passwort

Einloggen

Hast du noch keinen Account? [Jetzt registrieren.](#)

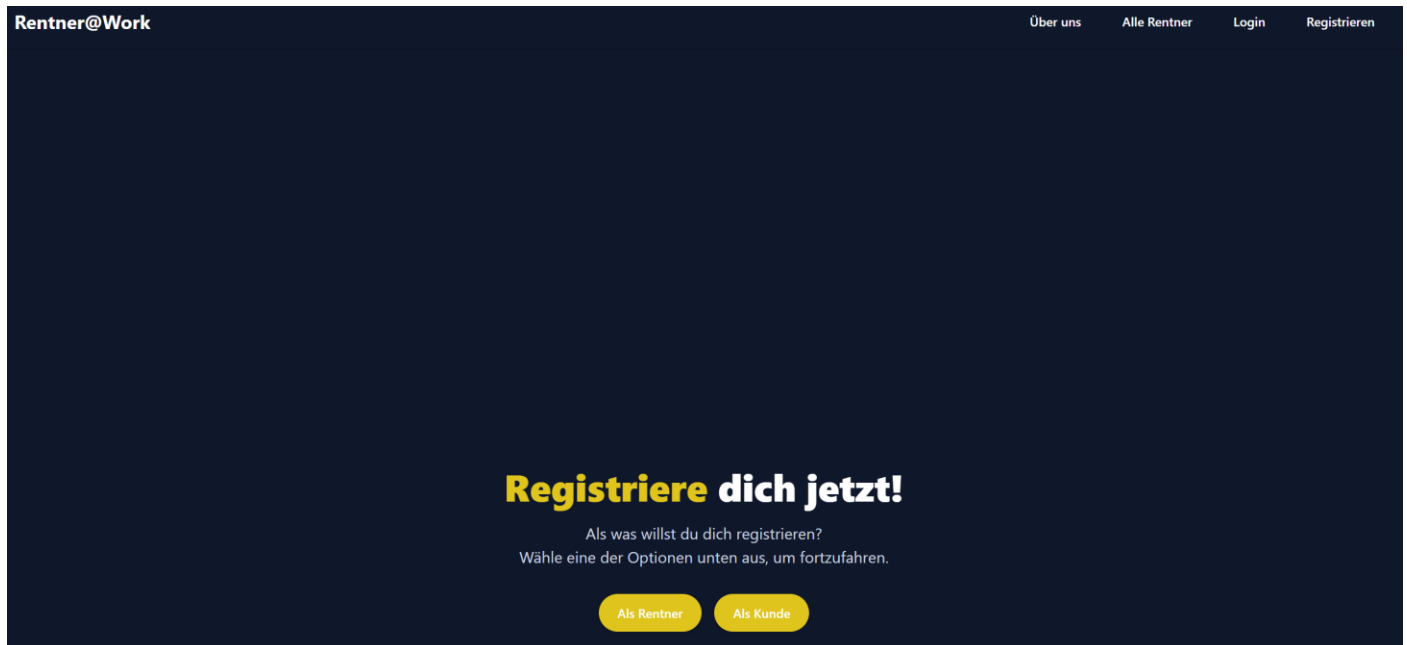
Diese Seite ermöglicht das Einloggen mit einem bestehenden Account. Nutzer können durch klicken auf den «Jetzt registrieren» Link auf die Seite Registrieren wechseln, falls noch kein Account erstellt worden ist. Durch Klicken auf den «Einloggen» Button wird bei einem erfolgreichen Login, das Login in die Session gespeichert.

Dateien:

- ../styles.css
- routes/login/+page.svelte
- routes/login/+page.server.js

2.5. Registrierung

Route: /register



Diese Seite ermöglicht es zu unterscheiden, ob man sich als ein Rentner oder als Kunde registrieren möchte. Beim Klicken auf den «Als Rentner» Button wird man auf die Seite /register/rentner weitergeleitet. Beim Klicken auf den «Als Kunde» Button wird man auf die Seite /register/client weitergeleitet.

Dateien:

- ../styles.css
- routes/register/+page.svelte

2.5.1. Registrierung Als Rentner

Route: /register/rentner

The screenshot shows the registration page for 'Rentner@Work'. The header includes the logo 'Rentner@Work' and navigation links: 'Über uns', 'Alle Rentner', 'Login', and 'Registrieren'. The main heading is 'Jetzt registrieren als **Rentner.in.**'. Below this is a registration form with the following fields: 'Firstname', 'Lastname', 'E-Mail', 'Passwort', 'Ort', and 'Dienste (z.B. Gartenarbeit, Haushalt)'. A yellow 'Registrieren' button is positioned below the 'Dienste' field. At the bottom of the form, there is a link: 'Du hast bereits einen Account? [Jetzt anmelden.](#)'

Diese Seite ermöglicht es zu unterscheiden, ob man sich als ein Rentner oder als Kunde registrieren möchte. Beim Klicken auf den «Als Rentner» Button wird man auf die Seite /register/rentner weitergeleitet. Beim Klicken auf den «Als Kunde» Button wird man auf die Seite /register/client weitergeleitet. Auch hier ist es möglich auf einen Link zu klicken, welches zur Anmeldung, bzw. Login-Seite führt, wenn bereits ein Account besteht.

Dateien:

- ../../styles.css
- routes/register/rentner/+page.svelte
- routes/register/rentner/page.server.js
- lib/utils/db.js
- bcrypt

2.5.2. Registrierung Als Kunde

Route: /register/client

The screenshot shows a dark blue web page for 'Rentner@Work'. At the top, there is a navigation bar with links: 'Über uns', 'Alle Rentner', 'Login', and 'Registrieren'. The main heading in the center is 'Jetzt registrieren als **Kunde**.' Below this heading is a registration form with five input fields: 'Firstname', 'Lastname', 'E-Mail', 'Passwort', and 'Ort'. A yellow 'Registrieren' button is positioned below the 'Ort' field. At the bottom of the form, there is a link: 'Du hast bereits einen Account? [Jetzt anmelden](#).'

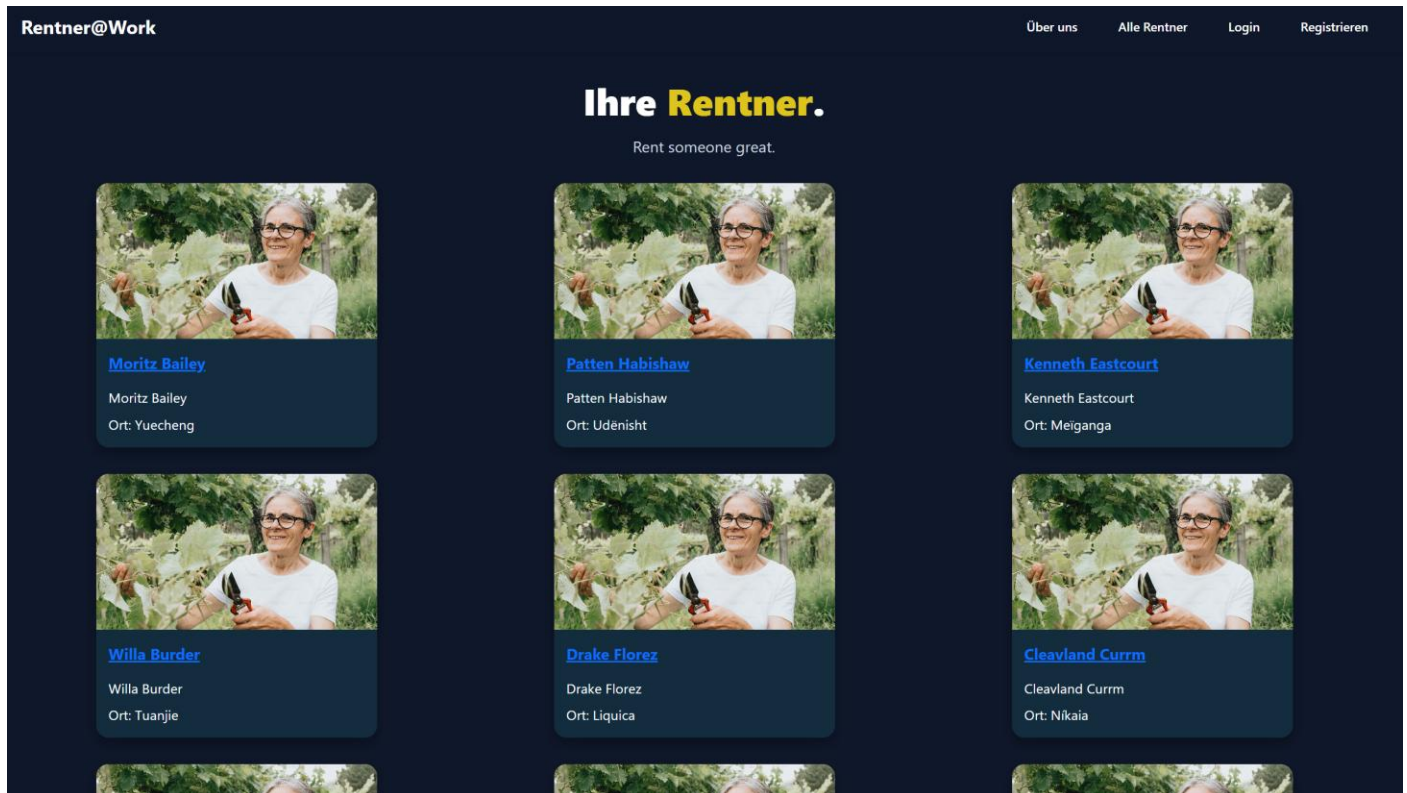
Diese Seite ermöglicht es zu unterscheiden, ob man sich als ein Rentner oder als Kunde registrieren möchte. Beim Klicken auf den «Als Rentner» Button wird man auf die Seite /register/rentner weitergeleitet. Beim Klicken auf den «Als Kunde» Button wird man auf die Seite /register/client weitergeleitet. Auch hier ist es möglich auf einen Link zu klicken, welches zur Anmeldung, bzw. Login-Seite führt, wenn bereits ein Account besteht.

Dateien:

- ../../styles.css
- routes/register/client/+page.svelte
- routes/register/client/page.server.js
- lib/utils/db.js
- bcrypt

2.6. Rentner

Route: /rentner



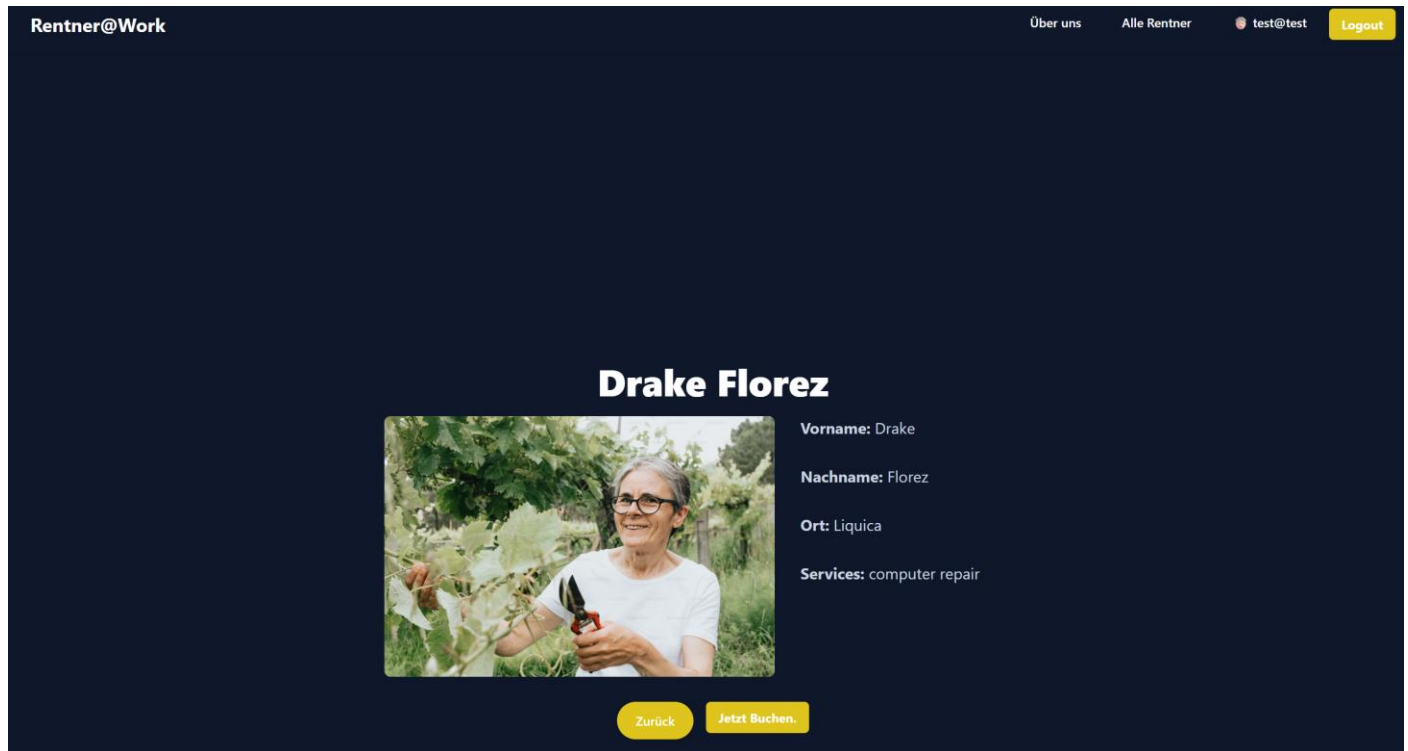
Auf dieser Seite werden alle Rentner aufgelistet. Die Auflistung erfolgt nach einem Aufbau, welches durch die `RentnerCard.svelte` definiert wurde. Beim Klicken auf den Link wird man auf die Seite `/rentner/[rentner_id]` weitergeleitet.

Dateien:

- `lib/components/RentnerCard.svelte`
- `routes/rentner/+page.svelte`
- `routes/rentner/+page.server.js`

2.7. Detailansicht der Rentner

Route: /rentner/[rentner_id]



Auf dieser Seite werden alle Rentner aufgelistet. Die Auflistung erfolgt nach einem Aufbau, welches durch die `RentnerCard.svelte` definiert wurde. Beim Klicken auf den Link wird man auf die Seite `/rentner/[rentner_id]` weitergeleitet. Beim Klicken auf den «Zurück» Button wird man auf die Seite `/rentner` weitergeleitet, bzw. zurück zur Gesamtübersicht aller Rentner. Beim Klicken auf den «Jetzt Buchen» Button wird man auf die Seite `/gebucht` weitergeleitet, bzw. zurück zur Gesamtübersicht aller Rentner. Die Buchung wird mittels einer MongoDB Query in der Funktion `createBooking(booking)` in der Datei `lib/db.js` realisiert und in die collection `bookings` gespeichert. Die Details des Rentners werden einer MongoDB-Query in der Funktion `findOne(params.rentner_id)` in der Datei `lib/db.js` realisiert.

Dateien:

- `Lib/utils/db.js`
- `lib/components/RentnerCard.svelte`
- `../..styles.css`
- `routes/rentner/[rentner_id]/+page.svelte`
- `routes/rentner/[rentner_id]/+page.server.js`

3. Erweiterungen

3.1. Login-Funktion mit anpassender Navigationsbar

Auf der Seite /login kann man sich anmelden. Die Navigationsbar wird je nach Login-Status angepasst. Auch die Button Auswahl auf der Startseite ändert sich. Dies wurde mit einem Formular und einer MongoDB-Query in der Funktion login(email, password) in der Datei lib/db.js realisiert. Die Login wurde selber gebaut.

Datei(en):

- lib/db.js
- routes/login/+page.svelte
- routes/login/+page.server.js

3.2. Registrierung-Funktion

Auf der Seite /register/client oder /register/rentner Seiten kann man sich registrieren und einen Account erstellen, mittels welchem man sich anschliessend auf der Login-Seite (/login) anmelden kann. Dies wurde mit einem Formular und einer MongoDB-Query in der Funktion createUser(user) in der Datei lib/db.js realisiert. Die Registrierung wurde selber gebaut.

Datei(en):

- lib/db.js
- routes/register/+page.svelte
- routes/register/client/+page.svelte
- routes/register/client/+page.server.js
- routes/register/rentner/+page.svelte
- routes/register/rentner/+page.server.js

3.3. Profil Löschen-Funktion mit anpassender Navigationsbar

Auf der Seite /profile kann man sein eigenes Profil wieder löschen. Die Navigationsbar dementsprechend auch angepasst und die Session beendet. Auch die Button Auswahl auf der Startseite ändert sich. Dies wurde mit einer MongoDB-Query in der Funktion deleteUser(userId) in der Datei lib/db.js realisiert.

Datei(en):

- lib/db.js
- routes/profile/+page.svelte
- routes/profile/+page.server.js