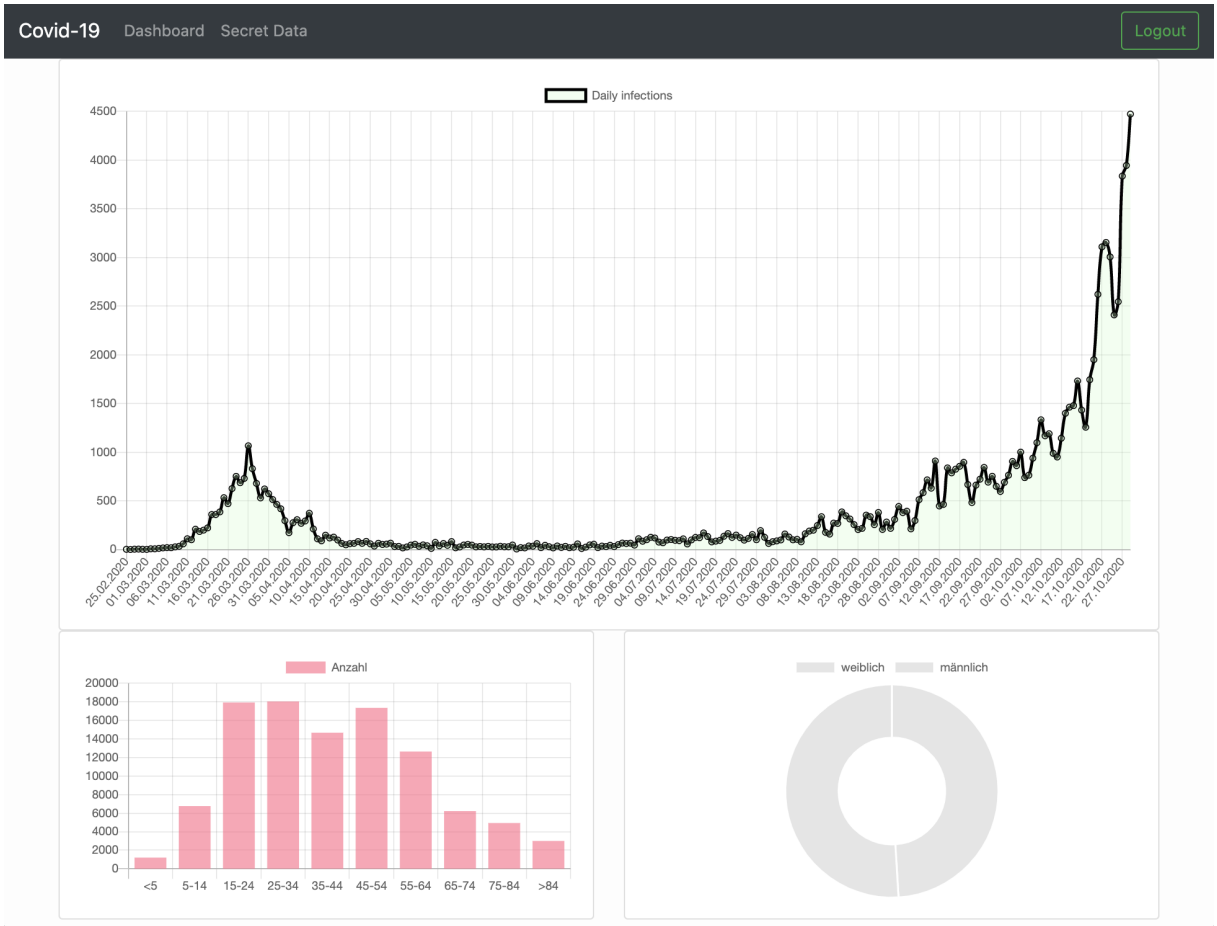
Covid-19 Dashboard

Implement a simple `dashboard` illustrating current numbers regarding `Covid-19 infections in Austria`. The dashboard shows graphics available for anybody but contains also a **secret data section** which visualizes sensible data about deceased persons. This section is protected by a password.



Implement at least `three charts` per section (public and secret). Decide yourself which data to display.

Task 1: Backend

Create a WebApi using **ASP.NET 6** for your backend system. Decide yourself which RESTful API routes to implement.

> You do not have to implement a database layer for this exercise.

The official data for Covid-19 infections can be downloaded from this URL as CSV-Files (read in the CSV file according to the data you plan to visualize):

https://covid19-dashboard.ages.at/data/data.zip

> You can see various charts using those numbers at this URL: https://covid19-dashboard.ages.at/

Task 2: Frontend - public

Implement the dashboard using `Bootstrap` and `charts.js` (https://www.chartjs.org/). Use the `ng2-charts` npm module in order to simplify using this library with Angular.

You can find a tutorial at this URL:

https://www.positronx.io/angular-chart-js-tutorial-with-ng2-charts-examples/

Create **separate Angular components** for the various chart types. Use the proposed module structure in your Angular project, create DTOs as you need them and implement a service to access the RESTful API.

Task 3: Frontend - secret

Secure the secret section using `guards`, `interceptors` and `JWT`. If the user selects the `secret data` route, check if he/she was authenticated already. If not, show a simple login page asking for the password.

> You can hardcode the password in the backend, e.g. `$ecret`.

If the user is authenticated, display a `logout` button in the navbar on the dashboard (see screenshot above).

Adapt the backend accordingly to support the JWT workflow.