#### General information

- You are to report the time spent on the project, so remember to keep a log of working hours.
- The project is to be carried out in groups of 4 students. Enroll for the groups at `https://sam.cs.lth.se/LabsEnrollSession?sessionId=37002&occasionId=834`. You can enroll on your own, the system will form groups automatically. If two or more students enroll at the same time, you will end up in the same group (the system will not split groups, but will add more students to existing groups)
- I leave the planing and organisation for the project work to you. I encourage pair programming and continuous discussions among the group members. Having responsibility of a component does not mean you have to do all the work. Help your group members, and they help you.

# Project description

The project is mandatory for the web programming course. The goal is for you to get hands on experience with a selected part of the course that has not been covered in the labs.
Mandatory parts for all projects:

- You are to develop a single page web application.
- Use a component based framework that renders the app at the client side. I can give support for react and angular, `https://angular.io` so this is my recommendation. It is possible to use other frameworks, for example Vue, but then I can not give support.
- The application should be slightly larger than the salad bar app from the labs and contain at least one HTML form.
- You must use a global state: context and reducer for react, a service for angular.
- Styling must be done using a package, prebuilt GUI components rather than using native HTML and CSS. In angular you can use material `https://material.angular.io`, or ng-bootstrap `https://ng-bootstrap.github.io/`. In react you can use `https://react-bootstrap.netlify.app`.

  This is different compared to the labs, where CSS classes were used to style the app. In the project, you are to get hands on experience with using prebuilt components and angular directives for styling.
- You must fetch data for your app from an external server. There are plenty of open data sources, for example: `https://www.dataportal.se`, `http://www.omdbapi.com`, `https://openlibrary.org/dev/docs/api/books`, `https://www.openstreetmap.org`. For angular you must use an angular service for this. For other frameworks you should use the recommended design pattern for that framework.

## Timeframe and deliverables

- 4/9 — hand-in of a project proposal. The hand-in is done in canvas. Only one in the group needs to hand in the proposal. Describe the functionality of your app and what packages you plan to use. The proposal can be half an A4 page, and definitely less than 2 A4 pages. I will give you feedback on the coverage of your proposal (to little/much work, to simple/complex task).

- 20/9 — deadline for final report. Hand-in is done in canvas. I believe in running code. I prefer codeSandbox, but a git repo also works. Hint, if you use github.com, you can create a codeSandbox that is synced with your repo, see `https://codesandbox.io/docs/git`. In addition to running code, you are to hand-in a written report, see bellow for details.

## The final report

The report should cover the following topics:

- Success ratio and lessons learned. Describe how much of your initial idea was actually implemented. You do not need to be detailed about what you got running, I see this from the running code. Rather, the focus should be on the parts that did not work out as intended.
- Main obstacles during the project. This can be anything, from group dynamics to getting a piece of code to run. (primarily so I can give advice and adjust the topics for coming years)
- A short reflection on which knowledges and skills you acquired in other courses that was most useful during your project.
- Individual statement of contributions. Here I expect that several group members have been involved in most parts of the project. Please also report the amount of time spent on the project. The reported time can be anonymous, i.e. student 1, student 2..., and will only be used to adjust the topics for coming years.
- The report is probably 1-3 A4 pages.

## Hints

- if you need a REST server you can use `https://www.npmjs.com/package/json-server`.

*Editor*: Per Andersson

*Contributors* in alphabetical order:
Ahmad Ghaemi
Alfred Åkesson
Anton Risberg Alaküla
Mattias Nordal
Oskar Damkjaer
Per Andersson

*Home*: `https://cs.lth.se/edaf90`

*Repo*: `https://github.com/lunduniversity/webprog`

This compendium is on-going work.
**Contributions are welcome!**
*Contact*: `per.andersson@cs.lth.se`