# 04.02.01. Web prototype

**Links:**
Login-page/
https://ingimar-eyfjord.github.io/04.02.01_web_prototype/Index.html
Login-page/
http://iesdesigner.eu/School%20Folder/04.02.01_web_prototype/Index.html

!ATTENTION! **** / There is one major problem with my site, it's the semantic code. So you might think it's funny like I do, but I've always been writing my code in  + ID tags. Which never got any issues from W3.org. But when I saw that we have to write semantically correct code I thought I had to change it up and use different names for the handles.

However I started using incorrect names for the tags like <modules> <schedule> and stuff like that, thinking it was semantically correct which it obviously isn't. Therefore the entire page is written incorrectly. I only realized this today (hand in day) I know what the requirements are, but I was hoping you can look past this part.

**Long term goal:** The long term goal is for students to have access to Fronter through their phone. To give the students a portable platform for their busy lives. This will in return increase the engagement from the student's and hasten their workflow.

Make a social media platform on fronter that will increase engagement between students and administrators.

**The problem:**
Currently the Fronter we use today is not mobile friendly, therefore a lot of students opt to wait until they reach their PC's in order to log on.

A lot of the problems caused by Fronter today is that nothing is responsive and the schedule is the least scalable thing there. It's almost impossible to use it through the phone.

**My solution:** Make the mobile version of Fronter easily navigable through design, designing an interface that looks and feels like an app is my approach to achieving this. Make a social media platform based on twitter with added functions like Direct Messaging system.

**Design:**

I started out by writing up and coding based on the design we used for the XD prototype. I designed using the same structure but ultimately I varied with design.

For the primary colours I decided to take the colours of the KEA page and use that for the header, navbar and the background the colour of the KEA logo is also used as an accent colour.

For the secondary colours I used the KEA style guide, I choose to use the light pastel colours. For the logos I got then from a website that has a lot of different logos, I chose a set of linear design logos.

**Functionalities & reasoning:**

**Login:**
There's a tutorial on YouTube that teaches how to do a login function, I might go ahead and implement that login function to my web prototype with a pre set up password and username.

**Modules:** I'm am essentially designing a drop down menu that directs the user between the semesters to their projects. The idea is simple enough but can be confusing if you don't really know the structure before hand, but then again, no one knew the structure to begin with anyway.

The idea is that here you will also be able to see your feedback from the teachers with an icon on the side of each project, the icon would make the element behave in the same way as it does when clicked. It will drop down (expand) with the current feedback.

The end path will lead the user to the current project.

**Current project:**
This is basically a shortcut. There is a way to this page from the homepage, and in the navigation bar in the bottom. Here students can easily access the current project instead of navigation through the modules.

**Newsfeed:** The newsfeed is meant to be a sort of social media platform with all the relative information that is circulating around KEA and the students, my inspiration was twitter. Here the administration and can post news articles and information as well as a "tweet" or a picture they want to share.
Inside the platform there's an opening box with more settings and functionalities (press on the picture or HOME button).
There you have information about yourself and your followers. The idea here is that you can search for other people, choose what hashtags to follow or unfollow.

**Schedule:**
The idea behind the schedule is derived from an app that I use for work, it's fairly simple to use and you can see the whole week in one viewport.

## The code: Login:

The login form has an input field, a type, name and a value fields. This form is used when you hook up a PHP database up with your site in order for users to log in, save their passwords etc.

### HTML

```
<loginform class="loginbox">
<img src="Icons/1x/Login-avatar.png"
class="loginavatar" width="512" height="512"
alt=""/>
<h1 style="color:white;text-align:
center">Login Here</h1>
<form>
<p>Username</p>
<input type="text" name=""
placeholder="Enter Username">
<p>Password</p>
<input type="password" name=""
placeholder="Enter Password">

<input id="login" type="submit" name=""
value="Login">

    <a href="#">Lost your password?</a><br>
    <a href="#">Don't have an account?</a>
</form>
    <img id="kealoginlogo"
    src="images/KEA_logo_EN_Web.jpg"
    width="1064" height="545" alt=""/>
    </loginform>
    </login>
```

THe login form is fairly simple, and the css code is as well.

The struggle I had at first was the <input> field, as the browser would like to input the information, but there was nothing hooked up to the website to catch it.

So I used Javascrtipt to overwrite/prevent the default function of it, and made it go to the homepage instead

### CSS

```
    z-index: 200;
    width: 80vw;
    height: 60vh;
    padding:7vh 2.5vh;
    background:#2B3449;
    top:50%;
    left: 50%;
    position: absolute;
    transform: translate(-50%, -50%);
    box-sizing: border-box;
}
.loginbox p{
    margin: 0;
    padding: 0;
    color:white;
}
.loginbox input{
    width: 100%;
    margin-bottom:2vh;}
.loginbox input[type="text"], |
input[type="password"]
{
    border:none;
    border-bottom: 1px solid #fff;
    background:transparent;
    outline:none;
    height:1.5em;
    color: gray;
    font-size: 1em;
}

.loginbox input[type="submit"]
{border:none;
    outline:none;
    height:1.5em;
background: #CA564A;
color:white;
    font-size: 1em;
    border-radius: 1em;}
```

### Javascript

```
document.getElementById("login").onclick =
function()
{event.preventDefault();window.location.href =
"04.02.01_web_prototype.html";}
```

## The code:Schedule:

A simple grid function here with a javascript that applies class to the elements, making them either display:none or display:block.

### HTML

```
<week-22 id="week22" class="week-22">
<schedule-grid class="schedule-grid">

<schedule-header class="schedule-header">
    <img class="backwardsbutton"
    src="Icons/1x/1x/backwards.png"
    width="212" height="472" alt=""/>
    <h2 class="header-info">WEEK 22, MAY-
    JUNE</h2>
    <h2 class="header-info">THEME:Basic
    UX</h2>
  <img onclick="onwardsweek23()"
  class="forwardsbutton"
  src="Icons/1x/1x/Forwards.png"  alt=""/>
      </schedule-header>

<monday class="schedule-item">
    <h2 class="schedule-info">TOPIC: Web
    prototype, test and pitch</h2>
    <h2 class="schedule-info">Animation
    principles with UI</h2>
</monday>
```

The schedule is nested in a div called <week-22> (semantically incorrect see top), <week-23> and so fourth.

Inside is the actual content that are then nested neatly in grids. This means I can copy paste the <week>'s and have them look the same.

### CSS

```
.schedule-grid{
display:grid;
grid-template-areas:
'header'
'monday'
'tuesday'
'wednsday'
'thirsday'
'friday'
'saturday'
'sunday';
margin-top: 10vh;
width: 100vw;
height: auto;
}
```

**Javascript**

```
function backwardsweek22()
{document.getElementById("week22").classList.togg
le("week-
22");document.getElementById("week23").setAttribu
te("class", "week-23")}

function onwardsweek23()
{document.getElementById("week22").classList.togg
le("week-
22");document.getElementById("week23").setAttribu
te("class", "")}

function onwardsweek24()
{document.getElementById("week23").classList.togg
le("week-
23");document.getElementById("week24").setAttribu
te("class", "")}

function backwardsweek23()
{document.getElementById("week23").classList.togg
le("week-
23");document.getElementById("week24").setAttribu
te("class", "week-24")}
```

The way to navigate the schedule is by using the two buttons that are on the schedule header, one goes backwards and the other forwards.
The onclick value on the html tag of the images triggers the function.
The javascript function then toggles the class elements of the <week> handles, between display:none and display:block

**The code: Newsfeed:**

Here I use a simple grid that aligns the newsfeed, inside are other grids that align the content relative to the contents size and text length. A simple onclick function reveals the box that comes from the left, it will change the class attribute from a display:none to a display:block with an animation coming in from the left.

**HTML**

```
<centerbox-newsfeed class="newsbox">

    <newsbox-one>
    <img class="newsbox-logo"
    src="images/KEA_logo_EN_Web-2.jpg"
    width="50" height="auto" style="border-
    radius: 50%;" alt=""/>
    <h2 class="newsbox-header-name">KEA</h2>
    <h2 class="newsbox-header">Husk at tømme
    dit skab senest 1. juli</h2>
    <p class="newsbox-paragraph">Kære
    studerende, alle lockers bliver tømt fra
    mandag d. 1. juli. Eventuelt glemt
    indhold med værdi opmagasineres hos KEA
    indtil 1. oktober 2019 og bliver derefter
    afleveret til politiets hittegodsk...
    </p>
    <img class="newsboximg"
    src="images/lockers.jpg" alt=""/>
    <h2 class="newsbox-hastag">#KEA-
    ADMIN</h2>
    </newsbox-one>

    <newsbox-two>
    <img class="newsbox-logo"
    src="images/twitteravatar.jpg.png"
     width="50" height="auto" style="border-
    radius: 50%;" alt=""/>
    <h2 class="newsbox-header-name">Ingimar
    Eyfjord Smarason</h2>
    <h2 class="newsbox-header">Sunset vibes!
    </h2>
```

Here you can see part of the structure for the newsfeed, the <newsbox> (semantically incorrect see top) has the content. And the CSS structures the content.

I realise now that with a class name on the <newsbox> handle I could have spared saved space and time and only use **one** CSS structure, instead of using one for each <newsbox>.

**CSS**

```
newsbox-one{
    grid-column: 2;
    display: grid;
    grid-template-areas:
        'logo name name'
        'logo textheader textheader'
        'img img img'
        'img img img'
        'img img img'
        'resttext resttext resttext'
        'hastag hastag hastag';
        padding-bottom: 20px;
        padding-left: 5vw;
        padding-right: 5vw;
        border-radius: 1vh;
        margin-right: 10vw;
        margin-bottom:20px;
        background-color: white;}
```

**The code: Module:**

I was using  CSS animation and JavaScript in order to make the user stay on one page, I'm am essentially designing a drop down menu. The end goal of this is to make the web prototype feel like an app and that you are actually still on the same page whilst looking for the information that you need.Doing this means that I hid some div's with CSS code and with an onclick function another CSS class is assigned which makes it appear and transform.

**HTML**

```
<projects id="second-semester-closed"
class="second-semester-closed">
    <a href="current-task.html">
    <h2 style="padding-top: 3.5vh;font-size:
    1em;">03.02: A Bicycle Guide for non-
    Danes</h2></a>
<modules id="modules-1">

<div id="module-1" class="module-1"><h2
style="padding-top: 3.5vh;color: #2B3449;">B.
Theme 01: Basic Web*</h2>
</div>

<div id="module-2" class="module-2">
<h2 style="padding-top: 3.5vh;color:
#2B3449;">C. Theme 02: Basic Animation*</h2>
</div>

<div id="module-3" class="module-3">
<h2 style="padding-top: 3.5vh;color:
#2B3449;">D. Theme 03: Basic Content</h2>
</div>

<div id="module-4" class="module-4">
    <h2 style="padding-top: 3.5vh;color:
    #2B3449;">E. Theme 04: Basic UX*</h2>
</div>
</modules>
```

The modules are structured in this way (semantically incorrect see top).

Elements that appear when the dropdown functions are activated are hidden by their default class name.

Using an onclick function they will have the .moduleoneopen css name and transform.

**CSS**

```
#modules-1{
    display: grid;
    grid-template-rows:
        5vh 10vh 5vh 10vh 5vh 10vh 5vh 10vh 5vh;}
.module-1{
    visibility: hidden;
    position: fixed;}
.module-2{
    visibility: hidden;
    position: fixed;}
.module-3{
    visibility: hidden;
    position: fixed;}
.module-4{
    visibility: hidden;
    position: fixed;}

.moduleoneopen{
    opacity: 0;
    grid-column: 1;
    grid-row-start: 2;
    grid-row-end: 2;
    z-index: 2;
    width: 70vw;
    justify-self:center;
    height: 10vh;
    background-color: #ECECEC;
    border-radius:1vh;
    text-align: center;
    justify-content: center;
    animation: openmodule .5s .15s forwards;}
.moduletwoopen{
    opacity: 0;
    grid-column: 1;
    grid-row-start: 4;
    grid-row-end: 4;
    z-index: 2;
```

**Javascript**

```
function FirstsemesterOpen()
    {document.getElementById("first-semester-
    closed").classList.toggle("first-semester-
    open");
    document.getElementById("module-
    1").setAttribute("class",
    "moduleoneopen");document.getElementById("mod
    ule-2").setAttribute("class",
    "moduletwoopen");document.getElementById("mod
    ule-3").setAttribute("class",
    "modulethreeopen");document.getElementById("m
    odule-4").setAttribute("class",
    "modulefouropen");}
```

Here we have the functions used for the onclick value.

**<u>What I couldn't achieve in time:</u>**

**Newsfeed:** I was wanted was a messaging function to be here (the time allotted was not enough to complete all of my wishes).
I also wanted to do a like function on each individual post as well as a commenting system and a repost button. This would be placed underneath the information on the right hand side.

**Schedules:**
The idea is if you press on the day it's supposed to drop down with more information about the day, lectures, lunch exetera.

**Navigation:**
The navigation is very app like in use. The navigation menu is on the bottom, the idea is to make a media query where it would display:none and display rather on the top with a regular font instead of icons. My workflow was making everything responsive from the beginning, so it's still responsive.

**Modules:** I did not have time to code the feedback solution. Neither did I have time to fix a bug in one of the drop down animation.

**Positioning Bugs:** There are a few positioning bugs on the page, the optimal viewing size is 375x812px. But otherwise the site is fully responsive.

**Semantically incorrect tag - fix.**
Since my website is made mostly with tags that are not semantically correct I have no choice but to hand it in like it is. Since I have no more time to fix it. Fixin it requires me to change both HTML and the CSS code and possibly the Javascript text as well.

**<u>TESTING:</u>**

**Trunk test test 1: Indre a MMD student.**
**Where you able to log-in?:"**Yes, very easy."

**Did you find the layout confusing?:"** No it was very clear and nice, liked it all, it was all clear."

**Did you find the hand-in?:"** Yes she found it. "

**Do you understand the newsfeed?: "**Yes, very clear and nice. But it's not customizable or what? I liked it. Very nice with the pictures simple and effective."

**Do you understand that there is a module path that lead to the current task and that there's a shortcut to the current task?:"** Not really. Mostly because the Icon was confusing. Was a bit lost in that sense, perhaps with using it for longer you will understand what's what."

**How was your UX?:"** Satisfactory. Despite the small bugs it makes sense, the colors and the picture might be to much, it was all clear but perhaps the icons for the current task and the module. Liked the little details."

"Overall I would very much like to use it in real life - main page is distracting at first impression (in the navbar). Visually very satisfying."

**Trunk test test 2: Patrik a MMD student.**
**Where you able to log-in?:"**Yes."

**Did you find the layout confusing?:"**No not really, found it very simple, and the design is effective in general."

**Did you find the hand-in?:"**Yes, it was fairly easy."

**Do you understand the newsfeed?: "**Yes, it was my the favorite part."

**Do you understand that there is a module path that lead to the current task and that there's a shortcut to the current task?:"**I didn't figure it out, it's a bit hard to see the difference, but if you know about it it's easier. It's written on the homepage, perhaps text underneath the icons in the navbar would help."

**How was your UX?:** "Really like the design, it's effective and simple to get around. Overall it's very consistent, the main page is also very interactive. And suitable for young audiences."

**Analysis of test:**
By analysing the test result gives me validation of the design principles used and tells me I was successful in solving the biggest problem Fronter has. The problem that students face is the messy interface of the current fronter. My prototype fixes those things and makes it more intuitive and quicker to access the information needed.

The only problem might be the structure of the modules. They could perhaps be a little more clearer. All in all I, along with my classmates feel that this is at least much clearer than the current Fronter we use today.

**What I take out of this project:**
The most interesting thing I learnt during this project was about PHP coding and database servers. Of course I did not have enough time to learn all of and implement PHP to my web prototype, but I really want it to have a login function.

The other thing I learnt about myself whilst doing this project was that I have a pretty decent grasp of JavaScript and CSS animation and HTML markup, I have had no trouble making the prototype looking like it looks and that makes me very happy.

I have to work smarter with CSS classes and ID tags, in order to save time and space.

And I think I will always remember from now on what semantically correct code is.

**Needed improvements on the web prototype:**
I need to fix the HTML handles so they will be semantically incorrect.
I need to fix positioning on a few elements as they tend to fall out/are displayed outside of their divs.

# Performance Test: Google Lighthouse