

The background of the slide is a grayscale aerial photograph of a dense urban area. In the foreground, there are several modern, multi-story buildings with a grid-like window pattern. A road with some traffic and a few people walking on the sidewalk is visible. The city extends into the distance under a clear sky.

Accessible Webtechnology - Introduction

Overview

The courses **Accessible Webtechnology** and **UX aspects of AAT** are connected.

The general idea of the courses is:

1. **UX Aspects**: create a mockup design (UI prototpye) for an individual app ✓
2. **Accessible Webtechnology**: create a functional prototype of the app
3. **UX Aspects**: do usability tests with the prototype
4. **Accessible Webtechnology**: improve the prototype based on usability test results
5. **Final presentation** of the results

Learning outcomes

After the course **Accessible Webtechnology** you:

- understand how web applications work
- understand the concepts of HTML, CSS and JavaScript
- can transfer visual mockups to a simple interactive web app prototype
- can create contents of a web app using HTML
- add simple interactivity to a web app using JavaScript and Vue.js
- understand how to style a web app using CSS
- know the most important aspects of an accessible web app
- can do basic accessibility tests of web apps using the NVDA or VoiceOver screen reader

Jump into web technologies



learning by doing a project...

Image generated by Google Gemini using this prompt:

generate an image where novice webdeveloper directly jumps into a pool of dangerous webtechnologies. The pool looks a little bit dangerous, but the webdeveloper jumps in with joy. The pool should contain many symbols and brands related to web technologies. The symbols should not be symbols of apps, but technologies like HTML, JS, CSS, Vue.js, React, Angular, ...

Project-based learning

- “ Project-based learning **integrates knowing and doing**. Students learn knowledge and elements of the core curriculum but also apply what they know to **solve authentic problems** and produce results that matter... ”
- “ ... it **refocuses education on the student, not the curriculum** – a shift mandated by the global world, which rewards intangible assets such as **drive, passion, creativity, empathy, and resilience**. These cannot be taught out of a textbook, but must be activated through experience. ”

Markham, T. (2011). Project-Based Learning. Teacher Librarian, 39(2), 38-42

Project goal

The final prototype we create should:

- give a **good idea how the final app works and looks like**
- provide **basic interactivity** for important functionalities
- have **good accessibility** and **work with screen readers**

We will **NOT**:

- create a fully functional app
- use any backend, database or integration with an external service

 **Important:** *Thinking and understanding* for yourself is more important than a great polished (AI generated) result!

Project goal example

Imagine you're creating a **prototype for a messenger app**. The prototype could be able to:

- select contacts
- display conversations
- input and "send" messages (not really sending)
- maybe some special features

Your app will **NOT** be able to:

- really interact with other users
- store messages on the internet / in a cloud service

Course Overview

Four lectures, each 180 minutes:

- **Lecture 1:**
 - basics of web-development
 - identify components of your webapp and transfer them to HTML
- **Lecture 2:**
 - introduction to Vue.js
 - add some interactivity to your app
- **Lecture 3:**
 - introduction to styling and responsiveness (CSS, Bootstrap)
 - adapt your app visually
- **Lecture 4:** improve your app based on the user evaluation (course *UX Aspects*)

Grading

The final grade results of **5 tasks** (80%) and a **final presentation** (20%).

The presentation is a **shared class** from courses *UX Aspects* and *Accessible Webtechnology* (~15min).

Tasks (80%)

- **Task 1:** 5% basic HTML components / views of your app
- **Task 2:** 5% add some interactivity to your app
- Task 3: style your app
- **Task 4:** 60% final submission (including improvements from user evaluation - *UX Aspects*)
- Task 5: 20% video (screencast) showing a main use case within your app using a screenreader

Mandatory: Task 1, 2, 4 and final presentation (each minimum 50%)

Task 3 is not graded individually, but it helps to **work step-by-step** in order to achieve the final submission (Task 4).

ECTS workload breakdown

Accessible webtechnology has **3 ECTS**, so about **75 hours** workload.

These consist of:

- **1h** preparations
- **14h attendance**
 - 4x 180 minutes main lectures
 - 1x 90 minutes final lecture for presentations (*shared with UX aspects*)
- **60h for tasks**
 - 54h for tasks 1-4 (implementing)
 - 2 hours screen reader video (Task 5)
 - 4 hours preparation for presentation