5G00DM03-3005/3006 Basics of Web Development, Course Assignment (max 60 XP) Instructions

Version History

- March 7, 2022: Version 1.0 published

1 General

Create a subdirectory with name `course_assignment` in your local Git working directory, and operate within it.

As the course assignment, you are to implement web pages handling weather data obtained via an API. The work is carried out individually and independently – using code written by another or sharing your code is strictly **prohibited**. (Suspected cases of fraud are going to be investigated. **Fraud leads to failing this assignment and thus the whole course as well as possibly to other disciplinary actions.**)

Your implementation must work at least with modern versions of Firefox and Chrome browsers. (The works are going to be tested mainly using Firefox.)

The Questions and answers discussion area in Moodle can be used to ask questions concerning the assignment. This is recommended especially, if you suspect that others might also be interested in getting an answer for your question. Also supportive Zoom meetings for discussing the assignment work details and opportunities to get support in TAMK premises are organized. (You can, of course, ask also **by email, but please, consider the options mentioned above first**.)

Deadline: if you want to avoid penalties for being late, you should submit your work successfully during **April 24, 2022 (EET)** or before. After that, until (and including) **May 4**, each time the date changes before your submission, 3 XP are reduced from the overall score. In other words, the cumulative XP penalty for being late is increased by 3 XP for each starting day (at mignight). Works submitted after May 4 but before the end of **May 15** are still going to be checked, but for these cases the maximum grade obtainable for the whole course is 1. **After May 15, submissions are not most probably going to be accepted anymore.**

2 Backend API

There is an example backend API available to be used as your data source:

- Latest 500 measurements:
  http://webapi19sa-1.course.tamk.cloud/v1/weather
- Latest 20 temperatures:
  http://webapi19sa-1.course.tamk.cloud/v1/weather/temperature
  - you can replace *temperature* by any other measurement type name
- Latest wind speed:
  http://webapi19sa-1.course.tamk.cloud/v1/weather/latest/wind_speed
  - you can replace *wind_speed* by any other measurement type name
- Wind direction, as hourly average values for the latest 24 hours:
  http://webapi19sa-1.course.tamk.cloud/v1/weather/wind_direction/23
  - you can replace *wind_direction* by any other measurement type name
  - you can replace *23* by any other integer value
- Latest active measurement type names (based on the latest 50 measurements):
  http://webapi19sa-1.course.tamk.cloud/v1/weather/names

3 Requirements for the Frontend Implementation

The actual work to be done as your course assignment is implementing a web page (or a collection of pages) that dynamically fetches weather data using the backend API and presents it to users. Towards this end, you are to implement index.html and possibly other HTML, JavaScript, and CSS files.

There are seven XP tiers to be reached (see Table 1). **In order to reach a tier, all the requirements for all the lower-numbered tiers must be satisfied as well as all the requirements of the tier itself**. (If the requirements of a tier are fulfilled only partially, the tier tier does not give any XP.)

Table 1: XP tiers and the corresponding requirements. You are going to get the number of XP associated with the highest-numbered tier you reach minus possible penalties.

Tier	XP for the assignment	Requirements (Remember: also the lower-tier requirements must always be fulfilled)
1	5 XP	• Consider the latest (i.e., most recent) 30 measurements fetched from the backend API. When index.html is opened with a browser, information is presented within an HTML page in tabular format. There are data rows corresponding to those 30 – and only 30 – measurements, and columns for at least these pieces of information:

		◦ row number (1–30), ◦ measurement time, ◦ measurement type, and ◦ measured value. The information presented in the table is correct (in accordance with the data got via the API). • The submission instructions have been followed carefully.
2	15 XP	• The user interface features (at least) these three views for navigation, each having some appropriate heading texts and containing data fetched from the API: • View 1: Shows information fulfilling the Tier 1 requirements. • View 2: shows the information concerning the latest 20 measurements of a single type (you can select, e.g., temperature measurements) in tabular format. • View 3: as View 2, but select another measurement type (e.g., wind speed). For Views 2 and 3, otherwise the same minimum requirements for columns apply than for View 1, but showing the (same) measurement type for each row is not required, if the type is otherwise clearly communicated to the user by the UI. The user can easily switch between the views using some kind of a menu or tabs. Only one view is visible at a time. (In this document, the views are referred as View 1, View 2, and View 3, but these should not be used as, e.g., actual heading texts. Instead, you should name your views in a more descriptive manner.)
3	25 XP	• Time information within the data tables is shown having the date and time columns separately. • Data tables have (correct) column headers. • Besides Views 1–3, the user interface also contains an *Info view* (reachable similarly as the other ones). This view should present some author information – at least • your name, • your email address, and • the course implementation you are participating in (3005 or 3006). (You are also free to document other information relating to the pages or their functionalities if you so wish.) • Views 2 and 3 present, in addition to the tables, also a bar chart

		visualizations (vertical bars, time increasing towards the right edge of the chart) of the data. You can use, e.g., the Chart.js library or the ECharts library.

4	35 XP	• Views 2 and 3 each feature a drop-down menu letting the user select a time interval to consider. The menu options are
		  • **Now**, the latest 20 measurements (i.e., Tier 2 defaults for these views), values are actual measured values,
		  • **24 hours**, latest 24 hours, values are hourly averages,
		  • **48 hours**, latest 48 hours, values are hourly averages,
		  • **72 hours**, latest 72 hours, values are hourly averages, and
		  • **1 week**, latest week, values are hourly averages.
		The data concerning the current observation interval selection are always successfully shown by the data tables and visualizations (after fetching them).

5	45 XP	• Besides Views 1–3 and the Info view, there is a fifth view into which the user can navigate.
		  • This view must contain a similar time interval menu as Views 2–3.
		  • There must also be another drop-down menu letting the user select the desired measurement type for this view. (At least rain, wind speed, wind direction, light, and temperature must be supported.)
		  • The data concerning the selected measurement type for the selected time interval is obtained using the API, presented in a table, and visualized using a **line chart**. (You can use, e.g., the Chart.js library or the ECharts library also for this.)
		  • For this view, the interval menu option "now" must show the latest **25** measurement values regarding the selected measurement type – or as many as it can, if there are less than 25 of this type within the latest 500 measurements.

6	55 XP	• The work is thoroughly styled to make your page visually appealing, clear, and nice and easy to use. The screen space is used efficiently. Some colors (other than browser defaults) or some images are used to improve the looks.
		  • If you use any images,
		  • they must either be created by yourself, used with appropriate permission, or be licensed so that you are free to use them (in which case the licence conditions must, of course, be met). Do not consider using graphics as a part of your pages as personal use. (Sometimes school works can be seen as such, but let us pretend now that you are actually going to publish the pages as a public service and need the corresponding rights.)

		• Document your rights to use the images on the Info view. Even if you have created them by yourself, tell also this being the case explicitly.
7	60 XP	• You have implemented an extra feature or functionality enhancing usability or usefulness. (This might be, e.g., some kind of a sorting or search functionality – or something else.) Document this as instructed in Section 4.

4 Documentation and Submission

Download the PDF form doc.pdf from Moodle and fill in the information concerning your work.

- If you want to get XP for Tier 7, you must document the respective feature or functionality to doc.pdf.
- **10 XP will be reduced for each of the mandatory fields (marked with asterisks) left empty.**

Commit your work and push the files into the main branch of the remote repository. Then, get the address to this commit of yours as usual (consult the lecture slides of Session 7 and the related video, if necessary). Paste this address to the correct field in doc.pdf.

Submit doc.pdf (containing now, besides other information, the information on your commit) in Moodle course assignment submission area.

N.B.: make sure the commit address is in the right format (ending with the commit SHA), and remember that you should document only ONE commit address! (It does not matter, if you have committed or pushed your files one by one or not; you should only provide information on the commit reflecting your repository in its final state – i.e., typically, the latest commit.)

5 Assessment

Your page should be usable simply by opening the local index.html file in a browser – setting up any server should not be required. Also, you cannot rely on any external libraries being available as local installations! All the basic functionality should be implemented with just the basic HTML, CSS, and JavaScript. Using external libraries (loaded from content delivery networks) is OK for drawing graphs, but generally, external dependencies should be used sparingly concerning this

assignment. (After all, this is a basic course meant to practice basic skills, not using some fancy libraries.)

The base XP for the assignment will be given based on the highest tier your work reaches – consult Table 1. Failing to follow the general instructions or instructions concerning submitting the work (e.g., using wrong directory or file names or providing several commit addresses) may result to XP penalties or the assignment not being checked at all (and being given 0 XP without further consideration). Minor bugs can lead to XP reductions. Submitting late affects the scoring of this assignment or the overall course grading as described in the Deadline paragraph of Section 1.

Fulfilling at least Tier 1 requirements is mandatory in order to be able to pass the course. (The total XP score gathered during the course must, of course, also be sufficient.)