Project work

Last update 3.10.2024

The aim of the course project is to build a simple application with HTML, CSS and JavaScript. In this course you are not allowed to use React or any other that big JS framework. All the tools presented in the course are of course completely ok.

There are three project options and you can select the one that you are most keen on. The max number of points from the project is 40 and 20 is required for a passed grade.

The project work is done individually and the submissions are done by submitting the project in CodeGrade (the project is manually graded though). The repository needs to include 1-2 page documentation PDF of the project: what was done, what tools were used and how many points the student would like to get from the project with justification for the points and of course user manual if the UI is not that clear.

The documentation MUST also include declaration of AI usage:

1) Name all AI systems that were used in the development of the contents of this document, and for each

2) How and where they were used (illustrations, proofreading, getting ideas for text, to generate diagrams etc.) or

3) Clearly state that no AI assistance or tools were used in this assignment.

This Declaration is a mandatory part of the submission and leaving it out means that the work is incomplete and should not be graded. Declaring that no AI was used in the development of the document, but getting a high number in TurnitIn AI check is also a valid grounds for failing the work just by itself.

Points generated from the course project

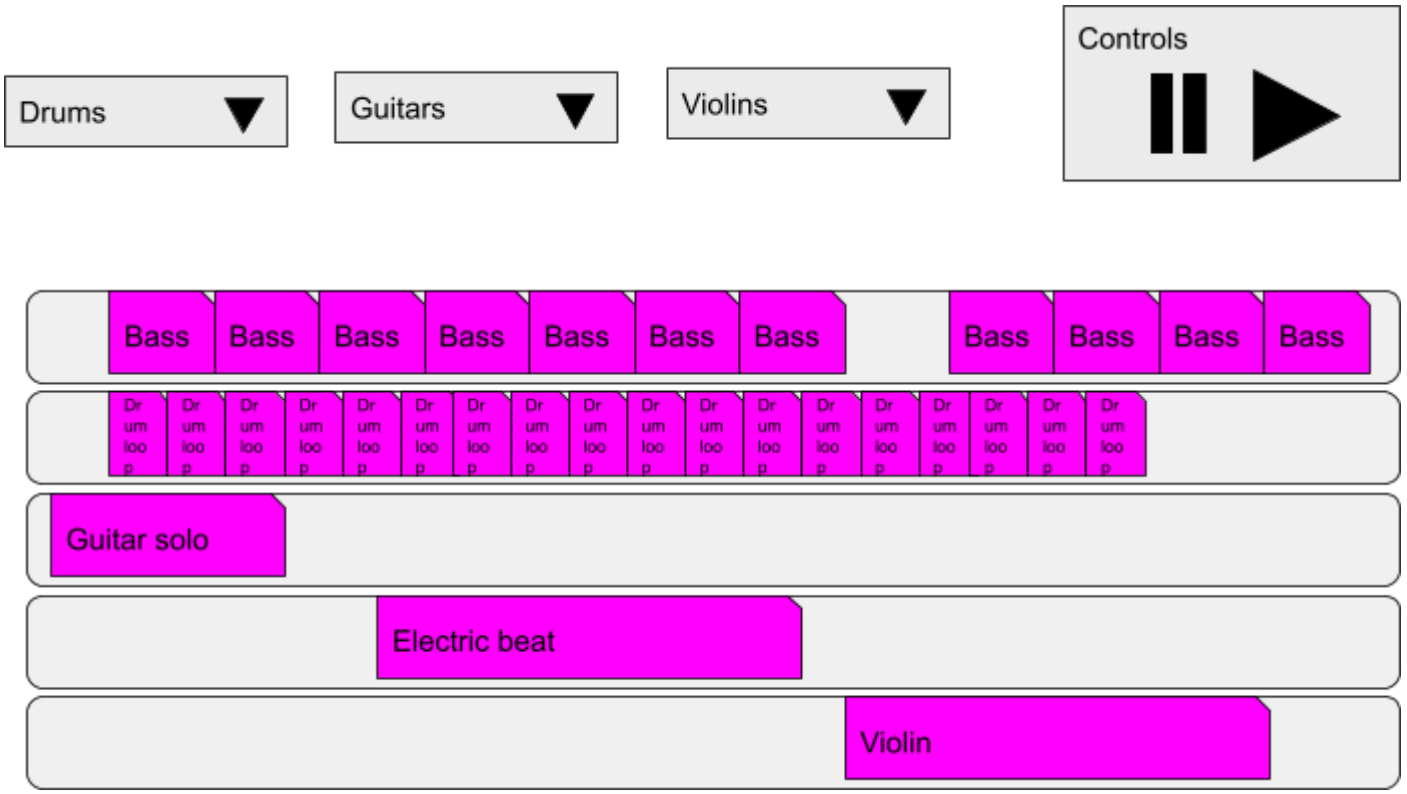
Feature	**Max points**
Well written PDF report	3
No report	-30
Application is responsive and can be used on both desktop and mobile environment	4
Application is not responsive	-2
Application works on Firefox, Safari, Edge and Chrome	3
The application has clear directory structure and everything is organized well	2

Application does not work	-30
CSS, JavaScript and HTML are all in the same file	-5
Inappropriate content, including hate speech -related memes and other trash	-100
<Your own feature: describe why should it be accepted>*	**n**

*Please be creative and add your own features as much as you want!

Project 1: Music maker

Continue the Music maker example from Week 2 and develop it further. Add CSS styling, support for infinite tracks, get the length of a sample and such. Make the best music maker there is and create new beats to conquer the world!



Feature	Max points
Drag'n'drop new instruments to the tracks (with mouse or touch screen)	4
Adjustable volume per track	2
Adjustable volume per instrument item	1
Able to download the final song	3
Istrument's length is visualized in the track	4
Available instruments are categorized (basses, guitars, drums…)	2
One can change the category of an instrument	1
Users can add as many tracks as they sees fit	1
Some tracks are looping tracks while some can be set to run only once	2
Use the web audio API and let users to record singing through microphone	3
User can delete tracks and instrument items	3
There are keyboard shortcuts to be used in several cases, like switch tracks/items and deleting them	2

Project 2: Statistic portal

Take open data from some portals (like Tilastokeskus https://statfin.stat.fi/PxWeb/pxweb/en/StatFin/) and use different visualization techniques to give a user additional information. Make sure that the application is usable with both desktop and mobile screens. You can for example try some / all of the following:

- Municipality elections
    - Show on map which party won in which municipality
- Visualize how the power of parties have changed over the years on some municipality (user can of course select the one)
- Visualize employment/unemployment rates on various places
    - Does that correlate with age distribution, educational level, or political parties?

Do the magic you really want to do with the data! Show how cool charts and maps it is possible to build with the tools introduced in the course or you can of course use some other tools if you see fit!

Feature	**Max points**
Drag'n'drop new data to charts/maps	4
The application show relevant data on a map and user has chance to change the data	3
The application show relevant data on a chart and user has a chance to change the data	3
User is able to switch between different layers of data on map	2
By clicking the map user has an option to get to additional charts covering that area	4
There are more than one item of data available (e.g. elections data, employment rate and number of residents) – this means that there are two API calls made	3
There are more than two items of data available (e.g. elections data, employment rate and number of residents) – this means that there are three API calls made	2
Data is combined and merged to generate new data, which is then visualized	3
Users can define what should be done to different data items (e.g. values are added, multiplied together etc. before visualization)	2
Able to download the visualization as a PNG (or SVG) image	2

Project 3: Weather app

Your task is to implement the best weather app there is! Use different forecast services and build an app that lets users see the weather forecast on the locations they want. You might get ideas from https://www.is.fi/supersaa/.

https://openweathermap.org/ is a good starting point.

NOTE: When you submit your project, make sure your API keys are working and then remove them after the project has been accepted.

Feature	**Max points**
User can search for locations	1
User can use his/her location GPS-coordinates (Geolocation API)	2
At least two data/forecast providers are used	3
At least three data/forecast providers are used	2
User sees the current weather at a specific location	1
User sees the forecast for the next 24 hour, hourly based	3
User sees the forecast for the next 7 days	3
All the weather forecast elements uses icons (and numbers) for e.g. sunny and cloudy weathers	3
The look and feel of the application reflects the current weather (e.g. it is blueish, when it is cold; reddish, when it is hot;, dark, when it is night…)	2
User sees simultaneously two forecast in a graph, e.g. there is temperature forecast for the next 24 hours and there are two lines telling how the data sources are providing (a bit) different data	3
User has the option to tag some locations as her favorites and thus access them from the favorites menu	2
User has an option to switch between celsius and fahrenheit degrees and kelvins	2