ITC 4214 – Internet Programming

Summer Term 2025

Instructor: Leonardos Mageiros

Polychroni Sophia

Midterm Project Report

22.6.2025

# Overview

*NoteTrack* is a client-side only, web application which aims to help musicians keep track of their musical ideas before they start composing them, so that their creative process is more organized and their composing faster.

# Description of task allocation system

* The users are able to add a task by adding a name, date, and description, and then submitting it.
* The tasks are saved to local storage and are displayed in a table for the users to view, edit, or delete.
* The tasks are dynamically counted into completed and pending.
* The tasks can be filtered and sorted.
* The tasks that are pending can be viewed in the welcome/home page.
* Music/ songwriting related tasks can be created in the studio page, and will also be added to local storage and displayed in the table when submitted. More specifically, in the studio the user adds a name, key, mood, instrumentation and any further notes/ideas, and the song task will be saved along with a proposed chord progression according to the key.

# Documentation of the coding decisions of tasks page, latest activity section and of studio page

## Tasks page

* Adding Tasks: Tasks are added via a form that captures the task name, due date, and description. Upon submission, the task is stored in the browser’s local storage as a JSON object. Each task is initialized with a default Pending status.
* Task Rendering: A dynamic table is used to display all tasks, allowing the user to filter by status (All, Pending, Completed) and sort by name, date, or status.
* Status Management: A <select> dropdown is included in each row to allow inline status updates. Changes saved to local storage.
* Task Editing: Tasks can be edited using a modal that appears when the user clicks "Edit". All edits are validated before being saved.
* Deletion: Tasks can be deleted individually, with confirmation prompts to prevent accidental removal.
* Metrics: Task summaries (total, completed, pending) are updated dynamically using a helper function that filters the task array.

## Latest activity

* Pulls all tasks from local storage and filters them to show only those with Pending status.
* These filtered tasks are then displayed.

## Studio

* Reuses the same logic and storage structure as the Tasks page to maintain consistency.
* When a user submits a new song idea via the form, the idea is stored as a task in the same local storage object.
* Dynamically updates the tasks table to include the new idea, treating it as a Pending song - task.

Overall

I used HTML, tailwind CSS (I chose it over Bootstrap because it is very similar with it in terms of use, however I liked more it’s minimal style, the design flexibility and the built in style variants like “dark:” for dark mode), feather icons for icons used, JavaScript and jQuery.

GitHub link

<https://github.com/1sooophiii/ITC4214_internet_programming_midterm_project>

# Personal reflection

Overall, I liked the whole development experience, and especially the fact that what I am developing and coding, I am able to directly view its results. I also liked finding an idea that I enjoyed and found interesting to implement and expand. I wish that I had committed more often during coding, however very often when I implemented something new, I sometimes forgot to add it. So, the last commits have more content than the first ones. I found easy implementing the dark mode using Tailwind CSS as it handles theme toggling with the dark class, and then all I had to do was to add the on click events to the buttons and save it to local storage. A bit more challenging was implementing the dynamic DOM updates, however selecting and manipulating DOM elements with jQuery feels more efficient, even though its syntax differs slightly from JavaScript, which was how I had initially started the implementation. Moreover, It took some time to work with objects that in localStorage are stored as strings; it was interesting to understand the JSON.stringify and JSON.parse that were required for it. JavaScript offers many useful methods for strings and arrays, like for sorting and filtering, but it required some research to find them and used them properly. Some future features I consider for my project is to add a MIDI export file of saved musical tasks to be downloaded and used in the user’s preferred DAW. To conclude, I am happy with the result of my project, yet I wish I had added some extra features discussed, but sadly I didn’t have the time to implement them in the end. Overall I believe I learned a lot in the process, and I hope that the final project will expand this idea even further and make it more interesting and complete.

*\*(Note: I will send the presentation during the upcoming week)*