UNIVERSITY of INFORMATION TECHNOLOGY and MANAGEMENT in Rzeszow, POLAND

Introduction to Web Technologies

# Final Project

Teacher:
Natialia Strukalo
Student:
Marcin Siebor w67069

## TABLE OF CONTENTS

## Description of topic, and objectives

The goal of my project was to create an online equivalent of Hobonichi Techo Planners. Hobonichi Techo Planners are a line of popular Japanese planners created by Hobonichi. The term “Techo” is short for “Technology” in Japanese and emphasizes the combination of traditional craftsmanship with modern design and functionality. Hobonichi Techo Planners feature unique page designs, such as gridded pages, monthly calendars, weekly spreads, and extra space for notes and sketches. The planner often includes special features for tracking weather, lunar phases, or other interesting details. Additionally, they may include articles, quotes, and illustrations to add some creativity to the planner. These planners developed and gained a devoted worldwide following including myself who tried to create an online equivalent for this project. The original objectives of this project were to have a home / landing page, a calendar view, a weekly view, and a daily view which would be divided in two to have a task management section and a note taking section. However, as time went on, the project proved increasingly difficult and was broken down to its two most essential elements in my opinion: the task management and note taking application. I decided to solve task management with a kanban board that would allow you to add, edit, drag-and-drop, and delete tasks by double clicking on them. This proved successful and I was able to accomplish it using local storage and JavaScript. The note taking application is similar to the kanban board as it also uses local storage and JavaScript. Finally, I wanted to host the website using AWS S3 which proved simple, however other AWS functionality had to be skipped due to lack of time on my part and wanting to complete the most important parts of the project first.

## Challenges and Problems Along the Way

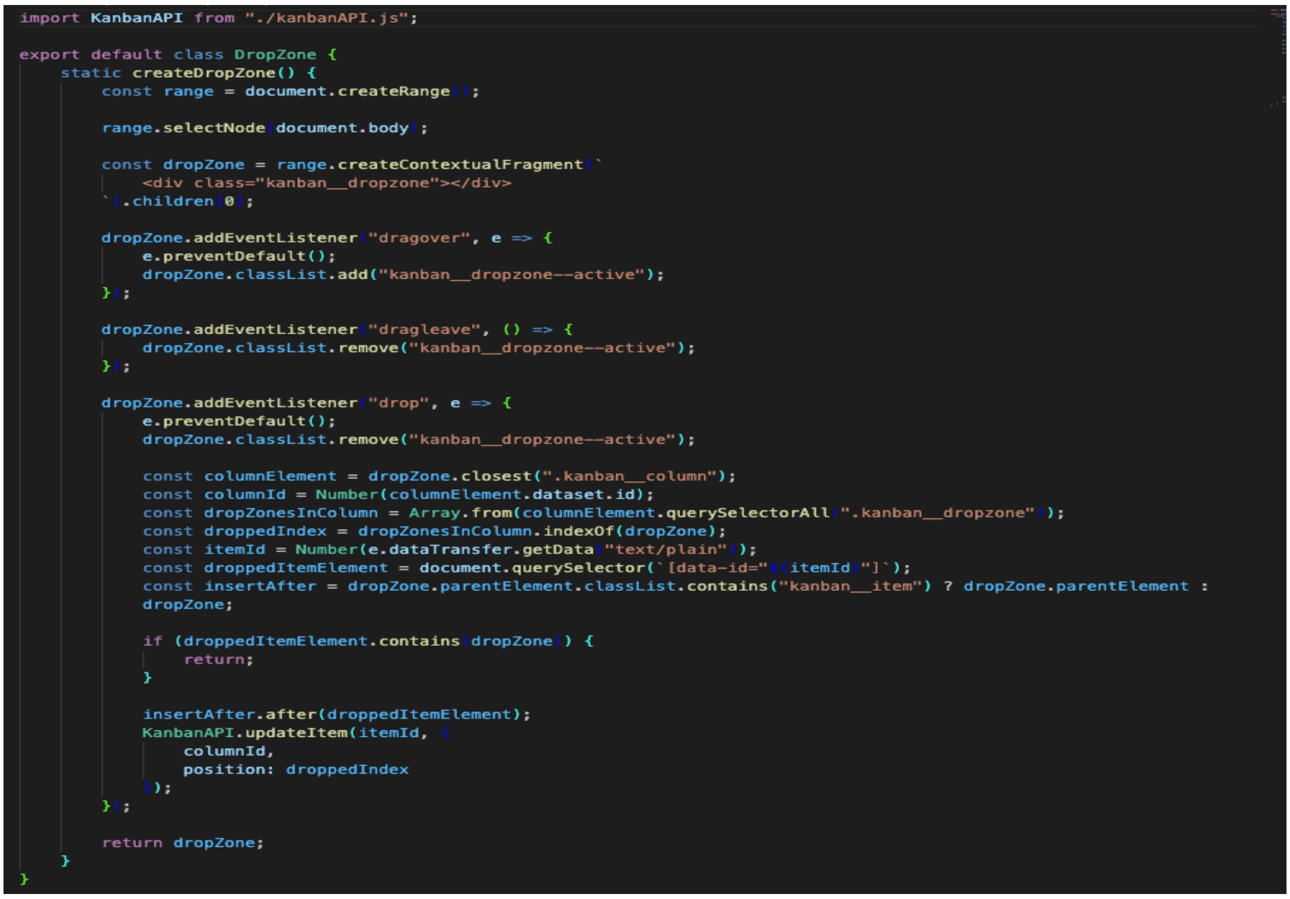
I had several challenges and problems as I worked on my project. The first was I was too ambitious and wanted to add too many features too fast which wasted time that could’ve been spent on more important parts of the project. One of the other challenges was implementing local storage with JavaScript to save notes and tasks on the website. Although I am happy how it turned out and satisfied with the experience which made me a better JavaScript developer, in retrospect, I wish I had used Amazon DynamoDB NoSQL Database instead as it likely would’ve saved me time and allowed me to polish other parts of my website or add some of the features I had wanted from the start. Otherwise, the project went by fairly smoothly. Going forward I’d like to make sure I start small with projects then add features and I’d like to work on my time management.

## DESCRIPTION OF THE TECHNOLOGIES USED LITERATURE

HTML5, also known as Hyper Text Markup Language, is used to define the structure of the website / webpage. I created four different HTML files for this project. My home page, index.html, is a simple page with its most complicated feature being the navigation bar. The same can be said about the About Page, about.html, which just explains basic features about the website. It also features a contact form to my email which I wanted to be functional using AWS Lambda, however, I ran out of time before I could implement them. The Notes and KanBan board also use very little HTML.

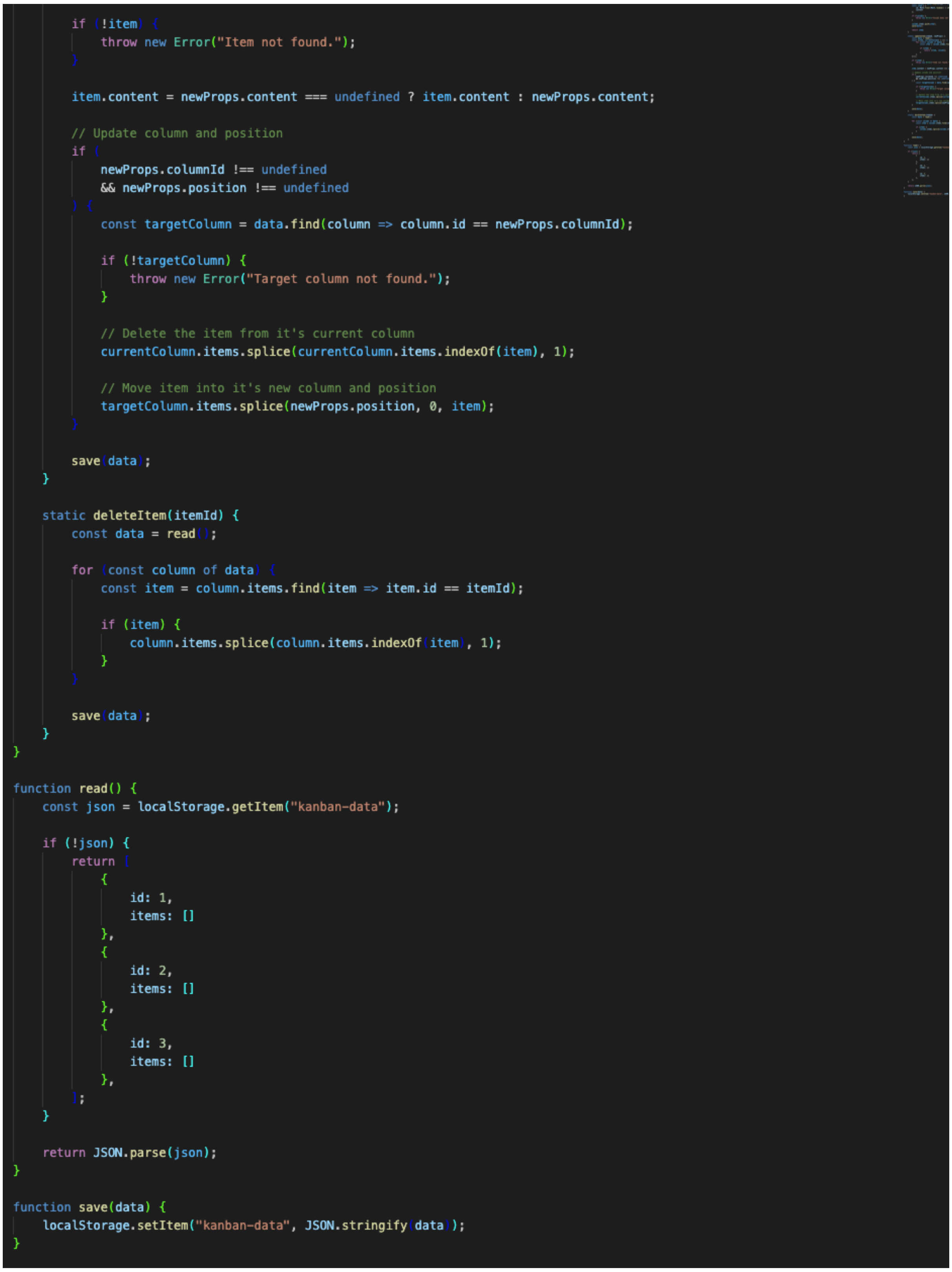
CSS3, also known as Cascading Style Sheets, is used to describe the presentation and visual appearance of HTML documents. I have a single CSS file, styles.css, that describes the presentation of my webpages. I chose a color palette of mostly white, black, and green.

JavaScript is a high-level programming language that enables dynamic and interactive behavior on web pages, allowing developers to create and control functionality such as interactivity, animations, form validation, and much more. I spent the majority of my time writing and organizing my JavaScript files for this project. The most difficult parts were easily the drag-and-drop functionality on the KanBan Board and the local storage for both the KanBan Board and Note Taking Application. Below should be the code for implementing the drop-zone for the KanBan Board.



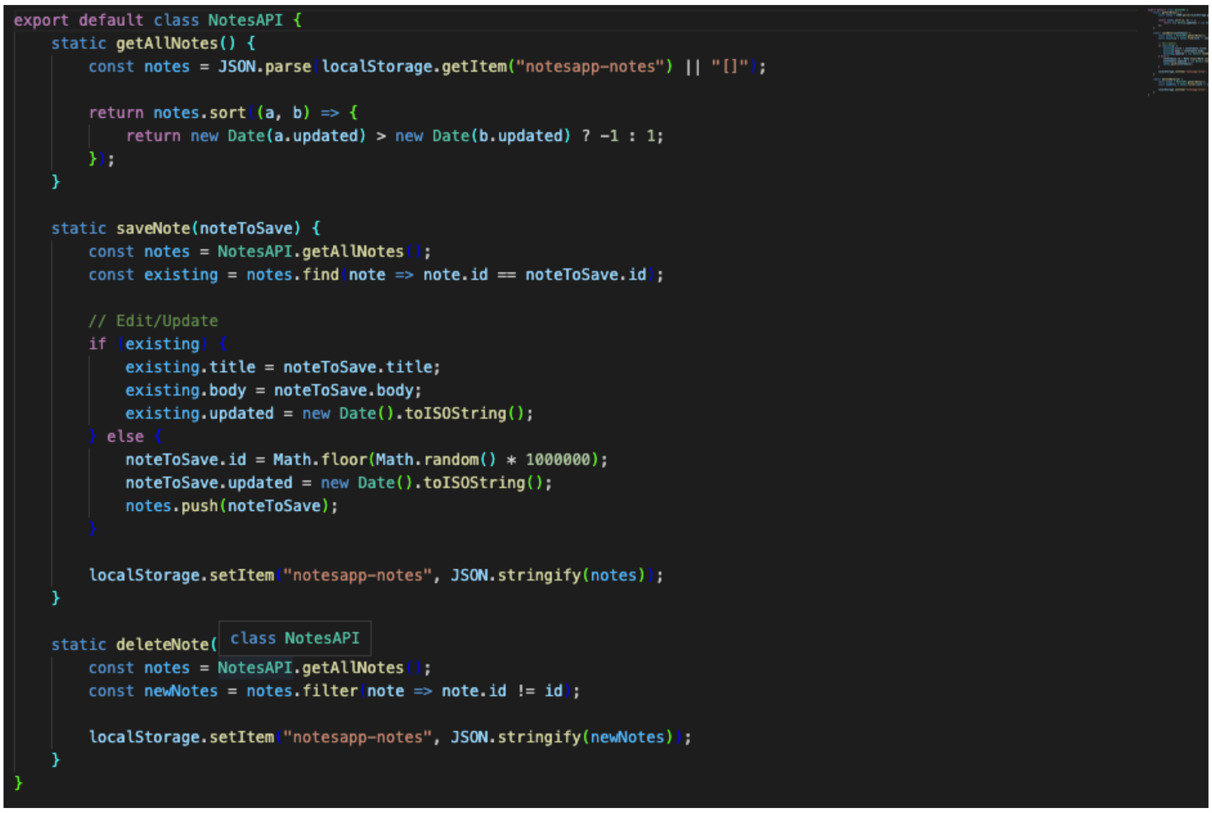
```
import KanbanAPI from "./kanbanAPI.js";

export default class DropZone {
    static createDropZone() {
        const range = document.createRange();

        range.selectNode(document.body);

        const dropZone = range.createContextualFragment(`
            <div class="kanban__dropzone"></div>
        `).children[0];

        dropZone.addEventListener("dragover", e => {
            e.preventDefault();
            dropZone.classList.add("kanban__dropzone--active");
        });

        dropZone.addEventListener("dragleave", () => {
            dropZone.classList.remove("kanban__dropzone--active");
        });

        dropZone.addEventListener("drop", e => {
            e.preventDefault();
            dropZone.classList.remove("kanban__dropzone--active");

            const columnElement = dropZone.closest(".kanban__column");
            const columnId = Number(columnElement.dataset.id);
            const dropZonesInColumn = Array.from(columnElement.querySelectorAll(".kanban__dropzone"));
            const droppedIndex = dropZonesInColumn.indexOf(dropZone);
            const itemId = Number(e.dataTransfer.getData("text/plain"));
            const droppedItemElement = document.querySelector(`[data-id="${itemId}"]`);
            const insertAfter = dropZone.parentElement.classList.contains("kanban__item") ? dropZone.parentElement :
            dropZone;

            if (droppedItemElement.contains(dropZone)) {
                return;
            }

            insertAfter.after(droppedItemElement);
            KanbanAPI.updateItem(itemId, {
                columnId,
                position: droppedIndex
            });
        });

        return dropZone;
    }
}
```

Below should be the JavaScript code for local storage on the KanBan Board.



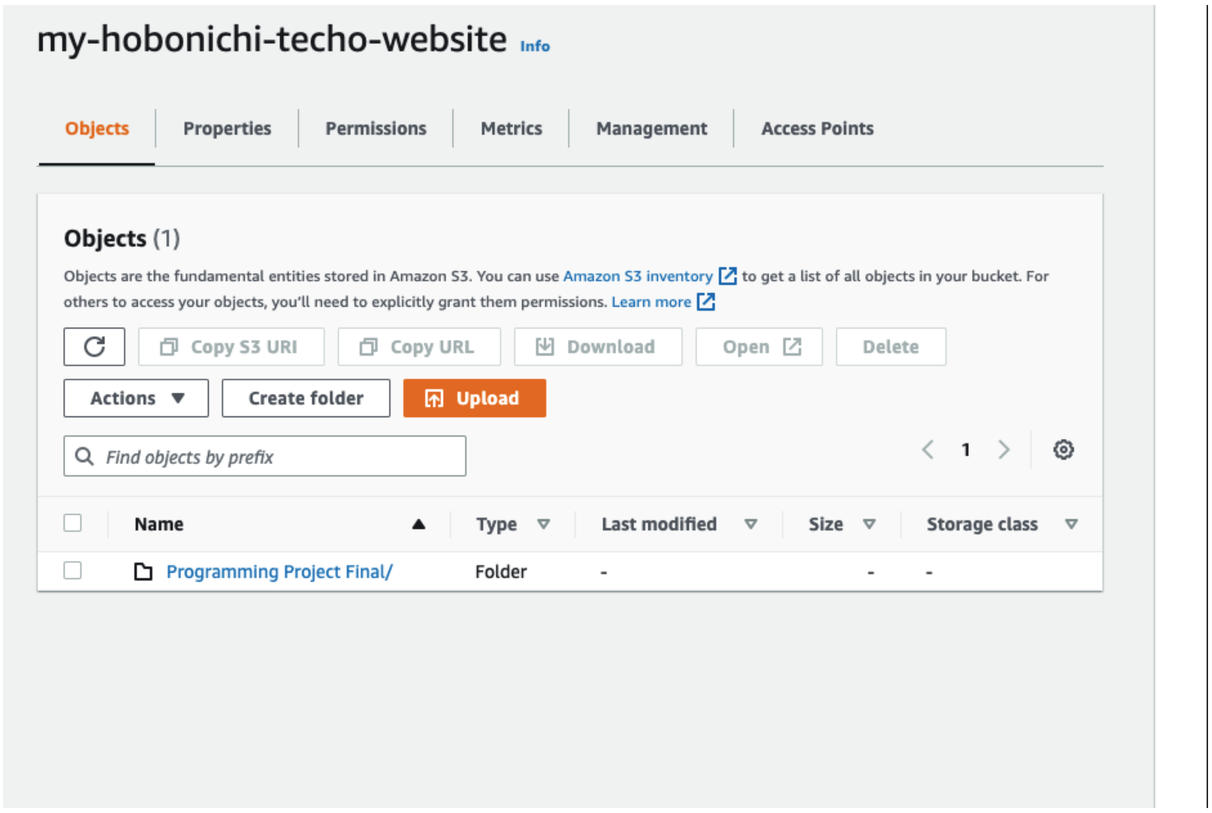
```
        if (!item) {
            throw new Error("Item not found.");
        }

        item.content = newProps.content === undefined ? item.content : newProps.content;

        // Update column and position
        if (
            newProps.columnId !== undefined
            && newProps.position !== undefined
        ) {
            const targetColumn = data.find(column => column.id == newProps.columnId);

            if (!targetColumn) {
                throw new Error("Target column not found.");
            }

            // Delete the item from it's current column
            currentColumn.items.splice(currentColumn.items.indexOf(item), 1);

            // Move item into it's new column and position
            targetColumn.items.splice(newProps.position, 0, item);
        }

        save(data);
    }

    static deleteItem(itemId) {
        const data = read();

        for (const column of data) {
            const item = column.items.find(item => item.id == itemId);

            if (item) {
                column.items.splice(column.items.indexOf(item), 1);
            }
        }

        save(data);
    }
}

function read() {
    const json = localStorage.getItem("kanban-data");

    if (!json) {
        return [
            {
                id: 1,
                items: []
            },
            {
                id: 2,
                items: []
            },
            {
                id: 3,
                items: []
            },
        ];
    }

    return JSON.parse(json);
}

function save(data) {
    localStorage.setItem("kanban-data", JSON.stringify(data));
}
```

Finally, the code below uses local storage for the note taking application.



```
export default class NotesAPI {
    static getAllNotes() {
        const notes = JSON.parse(localStorage.getItem("notesapp-notes") || "[]");

        return notes.sort((a, b) => {
            return new Date(a.updated) > new Date(b.updated) ? -1 : 1;
        });
    }

    static saveNote(noteToSave) {
        const notes = NotesAPI.getAllNotes();
        const existing = notes.find(note => note.id == noteToSave.id);

        // Edit/Update
        if (existing) {
            existing.title = noteToSave.title;
            existing.body = noteToSave.body;
            existing.updated = new Date().toISOString();
        } else {
            noteToSave.id = Math.floor(Math.random() * 1000000);
            noteToSave.updated = new Date().toISOString();
            notes.push(noteToSave);
        }

        localStorage.setItem("notesapp-notes", JSON.stringify(notes));
    }

    static deleteNote( class NotesAPI
        const notes = NotesAPI.getAllNotes();
        const newNotes = notes.filter(note => note.id != id);

        localStorage.setItem("notesapp-notes", JSON.stringify(newNotes));
    }
}
```

Amazon S3, short for Amazon Simple Storage Service, is a scalable and secure cloud storage service provided by Amazon Web Services (AWS) that allows users to store and retrieve large amounts of data over the internet. For this project, I chose to host my website using Amazon S3 as shown below.

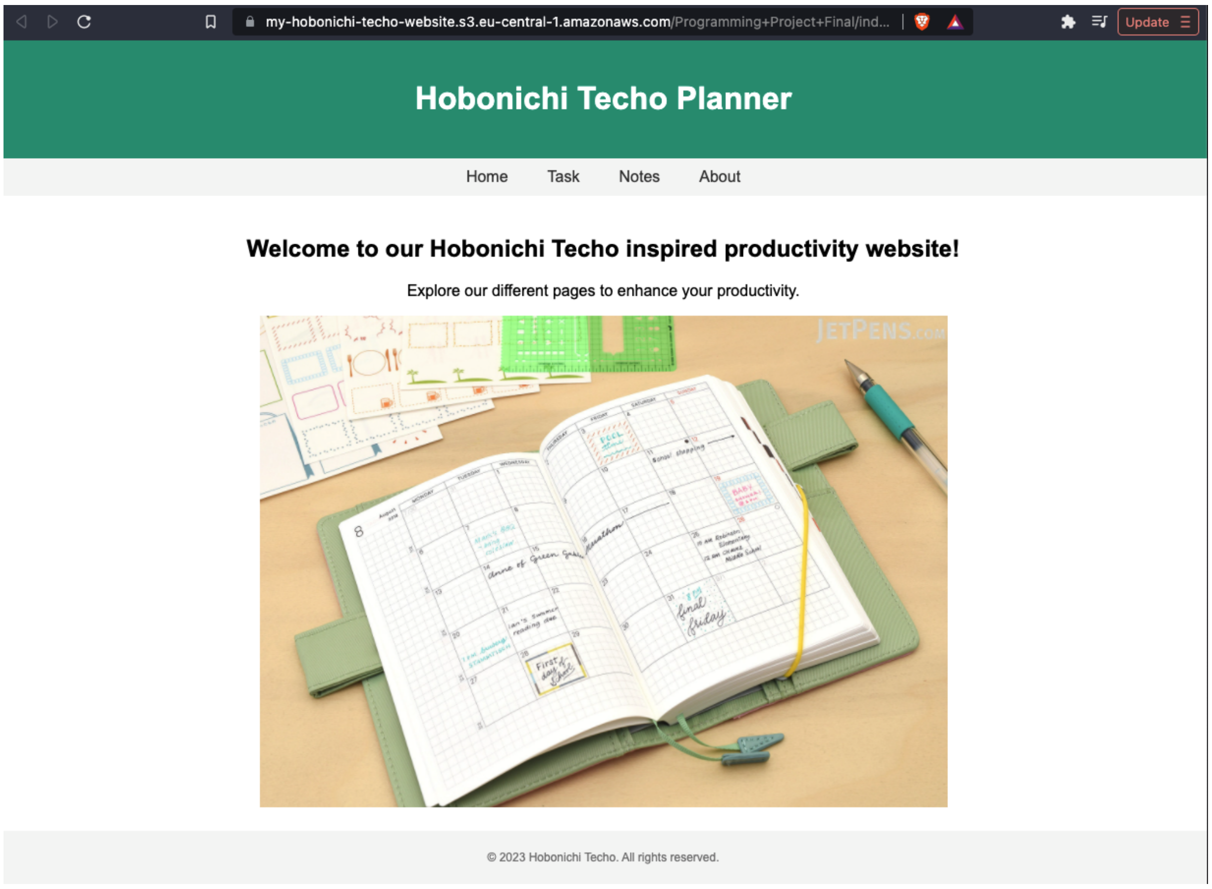


my-hobonichi-techo-website Info

Objects (1)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

| | Name | Type | Last modified | Size | Storage class |
|---|---|---|---|---|---|
| | Programming Project Final/ | Folder | - | - | - |


my-hobonichi-techo-website.s3.eu-central-1.amazonaws.com/Programming+Project+Final/ind...
Update

Hobonichi Techo Planner

Home Task Notes About

Welcome to our Hobonichi Techo inspired productivity website!

Explore our different pages to enhance your productivity.

© 2023 Hobonichi Techo. All rights reserved.

This should be the working link to my website:
[https://my-hobonichi-techo-website.s3.eu-central-1.amazonaws.com/Programming+Project+Final/index.html](https://my-hobonichi-techo-website.s3.eu-central-1.amazonaws.com/Programming+Project+Final/index.html)

Git is a distributed version control system that enables multiple developers to collaborate on projects, track changes to source code, and easily merge and manage different versions of files. I used Git to keep track of changes of my website so that I didn't accidentally delete it or make a change I regretted. I could easily go back to a previous version and continue working from there. I also uploaded the website on GitHub for the project.