BSc Computational and Data Science
Course: Computer Science (cds-205)
Instructor: Ana Petrus

Daily Dashboard

A web application for your daily needs

Oliver Schütz[1,*]

[1]*Fachhochschule Graubünden*
[*]*E-Mail: oliver.schuetz@stud.fhgr.ch*

January 5, 2025

Abstract

This paper intruduces the Daily Dashboard, a web application that aims to simplify the management of daily tasks. By providing a centralised platform for managing daily tasks, the Daily Dashboard helps users to stay organised throughout the day, while reducing time and complexity. This implementation uses a locally deployed database server, a public accessible frontend and user authentication . The paper concludes with other deployment options and functional enhancements.

GitHub Repository: `https://github.com/DotNaos/dashboard`

1 Introduction

Staying organised throughout the day can be a hassle, especially when time is sparse. There are already many solutions and tools available to help with this, but they are often scattered across different platforms and devices. This can make it difficult to keep track of everything and can lead to important tasks being forgotten or overlooked. But the biggest drawback is the inconvenience of investing these minutes in the first place. The Daily Dashboard should solve this problem by being one of the first things you see in the morning and the last thing you see before going to bed. Serving as the only piece of information you need, before taking on the day and customisable to your needs.

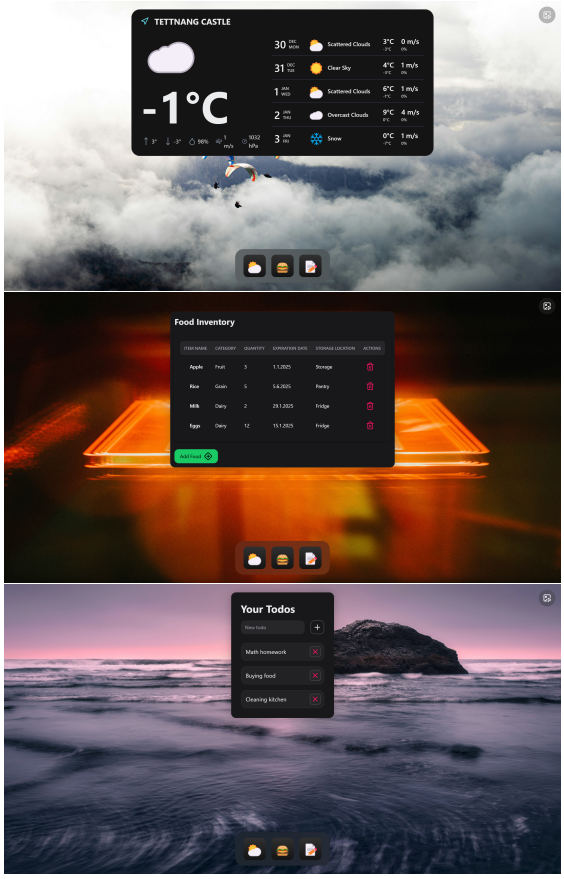

Figure 1: Website Pages

2 Methodology

2.1 Frontend

Using a website as the user interface offers the advantages of quick and efficient development, broad accessibility across devices, and extensive support for backend technologies. The frontend is built using a tech stack comprising React as the core framework, NextJS for server-side rendering and routing, TailwindCSS for consistent and efficient styling, and NextUI as the component library.

The website is organized into three main pages: the Weather page provides a quick forecast for the current and upcoming days, the Food Stock page displays a categorized list of food items with their expiration dates and storage locations, and the Todos page presents a simple to-do list for daily tasks (Figure 1).

Before accessing the main features, users must authenticate via a login page that supports both email and password registration as well as Google login. Upon successful authentication, users are directed to the main dashboard, where their personalized data is displayed (Figure 2).

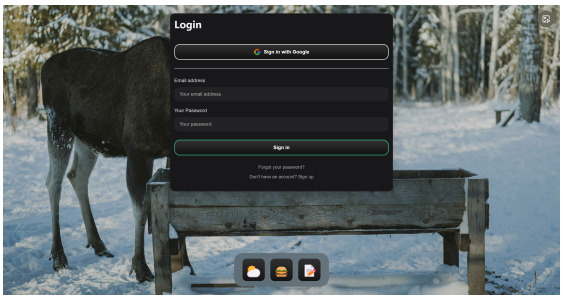

Figure 2: Login Screen

2.2 External APIs

Dynamic background images are retrieved from the Unsplash API and cached on the server side for one hour to maintain a fresh appearance without exceeding the free-tier limit of 50 requests per hour. Weather data, including the current conditions and a 5-day forecast, is sourced from the OpenWeatherMap API. This API offers a user-friendly interface for retrieving essential weather metrics such as temperature, humidity, and atmospheric pressure.

2.3 Backend

The website is hosted on Vercel at `https://cds-dashboard.vercel.app/`, which provides seamless deployment of NextJS applications stored in a GitHub repository. The application is automatically deployed upon each push to the main branch, making it a convenient solution for managing project updates.

User data, such as food inventory and to-dos, is stored in a Supabase database. Supabase, an open-source alternative to Firebase, supports both cloud and self-hosted deployments. While alternatives like MongoDB or traditional SQL databases exist, Supabase offers built-in authentication and API features that reduce setup complexity.

For local deployment, the Supabase Docker image can be pulled and run on any Docker-compatible device, such as a Raspberry Pi. Supabase provides a web interface for database management, and schemas can be imported and configured automatically via CSV files (Figure 3).

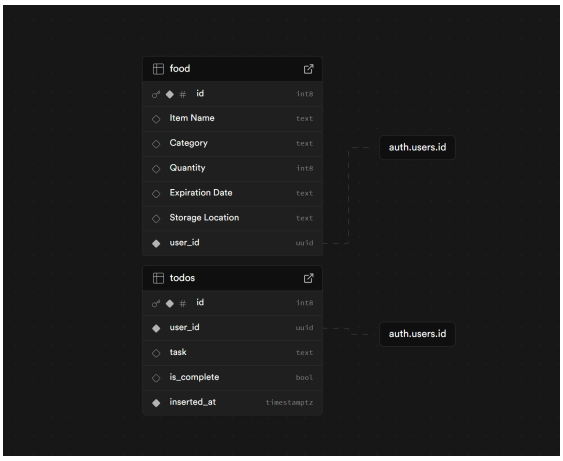

Figure 3: Database Schema

2.4 Tunneling

Since the Raspberry Pi is only accessible within the local network, port forwarding could be used, though it requires a static IP address, extensive configuration, and presents security risks (Wang, Xue, Feng, Zhou, & Mi, 2024) . A more secure solution is to use a tunneling service like ngrok, which establishes a secure external tunnel to the Raspberry Pi (*How does ngrok work? | ngrok documentation*, 2024) . Implementing ngrok involves installing the package on the Raspberry Pi, logging into an ngrok account, and running the command `ngrok http 3000`. The frontend and Google Cloud configuration URLs need to be updated to reflect the ngrok-provided URL to enable remote access.

3 Results

Implementing this was suprisingly simple, the google Login and authentication works out of the box. The Website is accessible from every device with a browser, however it is not made responsive, i.e. the ideal display size is tablet or desktop. The Weather is displayed

correctly, with correct data, but the location can differ from device types. During testing, on a desktop the location was always correct, but on a mobile device the location often switched to the nearest big city for example Zurich instead of Chur. Missing features are the ability to change the location manually, or edit the data for food stock and todos and automatically. Also the security of the website is missing many essential features, like database encryption, or rate limiting the api requests.

4 Discussion

Currently, the application must be opened manually on a phone, tablet, or PC. The next logical step would be to connect a touchscreen to the Raspberry Pi, enabling it to function as a standalone device. This would provide easier, continuous access to the dashboard without relying on external devices.

Another area of improvement is usability and user experience. Features such as GPS-based task autocompletion and integration with external applications like calendars or popular to-do list apps (e.g., Todoist) could make the dashboard a smarter, more centralized tool for users. Enhancing the user interface is also a priority, as the current interface is minimal and lacks features like editing, filtering, and sorting. Adding the option for users to take a photo of food items, with the application using image recognition to identify and automatically add them to the list, would significantly streamline inventory management. Additionally, implementing push notifications for expiring food items would further increase the dashboard's utility.

Beyond enhancing existing functionality, an entirely new feature I considered is an LLM-powered news feed that provides lecture summaries. This feature would run as a scheduled task on the server each morning, scraping course content from Moodle, summarizing it using a large language model (LLM), and displaying the summaries within the application. While a proof of concept has already been implemented, the feature was not included in the final project due to limitations in handling image-based and non-textual content within many current open-source LLM models. Therefore, this idea remains on hold until more advanced LLMs are accessible.

5 Conclusion

The Daily Dashboard simplifies daily organization by integrating weather updates, food stock, and to-do lists into a single platform. Future enhancements, such as a standalone touchscreen setup, smarter automation, and advanced features like LLM-powered summaries, will further improve usability and expand functionality, making it a useful everyday tool.

References

How does ngrok work? | ngrok documentation. (2024, December). Retrieved 2024-12-29, from https://ngrok.com/docs/how-ngrok-works/

Wang, H., Xue, Y., Feng, X., Zhou, C., & Mi, X. (2024, April). *Port Forwarding Services Are Forwarding Security Risks.* arXiv. Retrieved 2024-12-29, from http://arxiv.org/abs/2403.16060 (arXiv:2403.16060 [cs]) doi: 10.48550/arXiv.2403.16060