Working as a Software Developer

Work diary thesis

Divyansh -

Haaga-Helia University of Applied Sciences

Haaga-Helia Bachelor’s Degree

Diary Thesis

2024

Author(s)
Divyansh -

Degree
Bachelor of Business Information Technology

Title of the Report/Thesis
Working as a Software Developer, Working diary thesis

Number of pages and appendix pages
70

This diary documents my experience, learnings, and skills acquired while working as a software developer at Lahjakamu Oy from November 25th to January 20th. Throughout this job, I had the opportunity to learn various things, including daily tasks, the working environment, professional culture, organizational goals, weekly evaluations, and discussions.

I have shared all of these aspects in my thesis, along with my professional growth. Additionally, I've explored topics such as managing work-life balance, improving professional communication, collaborating within a team, and reflecting on my personal experiences as a software developer.

I strongly believe that sharing this experience will help others gain a deeper understanding of the software development profession and the associated working culture and lifestyle. Overcoming challenges has become an integral part of my daily routine, showcasing my strength and efficient problem-solving abilities. This experience has reinforced my commitment to continuous learning and gaining more hands-on experience in the software field.

Keywords
Software development, frontend, backend, JavaScript, React JS, Express

Abstract

Contents

1 Introduction

My topic is “diary thesis – working as a developer” and I chose this as I have relevant work experience by working at Lahjakamu Oy. The name of the employer is Lahjakamu Oy, and their work is providing personalized gift cards for various stores.

The employer is Lahjakamu Oy, a company dedicated to offering personalized gift cards for a diverse range of stores. The work environment at Lahjakamu Oy is dynamic and innovative, aligning with the evolving nature of the technology and e-commerce sectors.

In my role as a software developer, my primary responsibilities include designing, coding, testing, and maintaining software systems that facilitate the creation and distribution of personalized gift cards. Additionally, I actively try to collaborate with cross-functional teams to ensure the integration of software solutions.

I have been actively performing my current job tasks at Lahjakamu Oy, from 25th November gaining valuable knowledgeable into the company's operations and contributing to the development and improving their software solutions.

The expertise demanded by my job tasks consists of being proficient in programming languages such as React JS, Java, Python, etc. knowledge of software development methodologies, and a solid understanding of e-commerce and personalized solutions. It also requires being good with communication, planning project and taking leadership along with setting clear goals and deadlines. I feel adaptability and a continuous learning mindset are crucial in keeping updated with the technology.

I feel my skills developed as I learn more about the programming world and be an improved developer day by day. The experience I got from working at Lahjakamu is different and make me learnt how to work in professional manner.

From a sustainability perspective, the thesis work aims to explore how software development practices can contribute to the efficiency and eco-friendliness of personalized gift card solutions. This involves considering the environmental impact of software development processes and proposing sustainable practices.

Table 1. Overlay matrix of the in-text connections in the diary-based thesis

Own professional development objective	Theoretical framework	Observation week	Results of own professional development
Objective 1	e.g., 2.4, 2.6	e.g., weeks 2,3,8	e.g., 3.2, 3.3, 3.8 ja 4.3

Own professional development objective	Theoretical framework	Observation week	Results of own professional development
Objective 2	e.g., 2.1, 2.3, 2.5	e.g., weeks 1,10	e.g., 3.1, 3.10 and 4.5
and so on	and so on	and so on	and so on

2 Description of the initial situation

The next chapter in the report is a description of the initial situation that analyses the current work, stakeholders, and workplace interaction situations.

2.1 Analysis of your current work

My specific tasks vary from day to day, depending on my goals. Most of the times its about developing full stack applications using technologies such as MERN stack Mongo DB, Express, React and Node JS. Most of the times the work is in JavaScript. It also includes learning new technologies such as Maria DB, AWS, and more features such as private routing in react. I use visual studio code as my code editor and GitHub for code management. It also includes testing backend API routes with unit tests with JEST. Various tools such as Git, GitHub, Bitbucket and Atlassian Jira are used for project management and version control.

In order to excel with this job, one needs strong proficiency in JavaScript, Typescript, Java, HTML, CSS, NODE JS, Express JS and knowledge of databases such as Mongo DB, PostgreSQL. Knowledge of good coding practices, folder structure, clean and organized code are also necessary over long period of time to make code more manageable. One should also be familiar with REST API development as well as how to integrate it into the project, cloud platforms like AWS and Google Cloud platform.

My necessary know how has been acquired from my formal education which is bachelor's degree in business information technology where I'm specializing in Software Engineering. Along with that my work experience as a Software developer has taught me a lot of work knowledge and self-learning. I have also been doing personal projects where I did a lot of experimentation, adaption, and self-learning.

The stage I reached in professional development is mid-stage or Skilled performer as I have gained practical experience in full stack development, knowledge of how working life as a software developer is and a lot of practical or hands on experience.

My investments for the future would be to continue learning and staying updated about new technologies, best practices and develop expertise in new AI related technologies along with cloud computing. I also improved on my soft skills such as leadership, self-initiative, and teamwork. I'll be taking on more responsibilities. I aim to become a mid-software engineer within the next 2-3 years.

I do occasionally seek guidance from team lead when stuck on issues while working on ticket but that has not come in my way to still be independent on most of tasks. I actively contribute to project

development and guide others even though I don’t have that level of experience yet. I still have quite lot to learn but looking back to where I was at the start, I have improved x5 times. I’m capable of assisting others on technologies which I’ve worked on and having solved countless tasks.

2.2 Stakeholders

The stakeholders in my company are the customers, partner companies, developers, manager, and upper management. The stakeholder which are central and of upmost importance are the customers as the things I work on directly affects with the customer experience.

The management takes decisions on how the things should be or what changes are needed which is then implemented by me and the development team. At last, the customer gets to experience them.

2.3 Interaction situations

The working style in my team is scrum, we used scrum methodology starting with Sprint planning and then regular daily sprint meeting 10-15 minutes to keep everyone updated about the project and what they have been up to, including their today’s goals, at the end of sprint we had sprint retrospective and sprint review. This all helped in having frequent communication with meetings and slack to ensure tasks are being aligned with the goal.

In the sprint review (end of sprint) accomplishments were discussed while the stakeholders gave everyone feedback which helped to acknowledge the things that could’ve been improved. Then sprint retrospective was held where improvement areas are identified along with plan and set goals for an improved sprint ahead.

The daily meeting also assisted in opening about challenges and seeking help if required. After all the work-related discussions are out of the way there are casual conversations which enables more positive, friendlier, and comforting workplace for everyone. It helps everyone to adjust to the team.

3 Diary entries

I have been working at Lahjakamu from 25th November 2023 and currently as I write this diary I'm still working regularly at the company.

3.1 Observation week (27.11.2024 - 03.12.2024)

Monday 27th November

This was my first week of working at lahjakamu. It was a lot to consume but it went like this that first I had discussions with the development team and managers how the work is done. I got to know the team and discussed with them about the work life here. I had my onboarding session where I went through the company policies, procedures and do all the necessary paperwork. All this wasn't the most exciting part of it but I understand that it was crucial from a company's perspective.

But the real exciting thing was setting up my development environment while following the company's guidelines. I installed the necessary node version, setting up vite bundler and creating the project from scratch. It really felt satisfactory doing so much within just one day, but it was a great learning experience.

The extra thing I understood was that this job won't be just about coding or programming but also about continuous learning, teamwork, the connections I'll be making while working there and understanding the work culture combined with productivity. It makes you feel that you actually belong there and that you are actually part of a team, after a while you are well settled into the company.

It was a lot to take in during the first day, that also included solving errors, figuring out solutions to issues. Thanks to bit of googling which helped me resolve the issues. I felt super thankful for the supportive team as well.

I finally could start to feel how it is to work professionally within a company or how professionals work.

Tuesday 28th November

During my daily sprint, I engaged in a detailed discussion with my colleagues about the progress I had made. We explored how everything was going for me, addressing any issues I was facing. One particular challenge I encountered was an error message that stated "semver is not a function." To tackle this problem, I sought assistance from the tech lead, and we embarked on a trial-and-error process, attempting various solutions. We tried reinstalling the project, downloading necessary libraries, and even searching for solutions online. After investing around two hours in these efforts, we finally found success by deleting a node_modules folder in my C drive. Though I didn't grasp why this action resolved the issue, it allowed the program to run, enabling me to take on more tasks.

Recognizing the importance of sharing knowledge, I decided to document the issue I faced so that others who encounter the same problem in the future can save time and resolve it quickly. Sometimes, solutions to problems can be unexpected and divergent from what one might anticipate, requiring a broader perspective and a willingness to learn something new.

These experiences not only aid in expanding my knowledge and skills but also contribute to my growth as a developer. By continually seeking solutions and exploring different approaches, I am gradually transforming into a more accomplished and proficient professional in my field.

Wednesday 29th November

On Wednesday, the day started off simple. I began by sharing updates on what I had been up to, and soon after, I received news that I would be taking on a new project. As I got to know about the project, I discovered that I would need to familiarize myself with meteorJS, a relatively new JavaScript web framework written in Node JS. It was both exciting and challenging to get a strong understanding into a framework that enabled rapid prototyping and the creation of code that could function across multiple platforms.

Additionally, I came across a tool called NVM which was incredibly useful when it came to managing different versions of Node JS. It ensured that everyone working on the project had the same node version, which helped in collaboration. Exploring nvm introduced me to various commands, such as "nvm ls" (to list all the installed node versions), "nvm install 12" (to install version 12 of node), and "nvm use 17" (to switch to node version 17). These commands allowed me to navigate the project efficiently.

In addition to familiarizing myself with meteorJS and nvm, I took the necessary steps to get the project up and running. To begin, I cloned the project from GitHub, ensuring that I had all the latest code and updates. Next, I downloaded the required libraries using npm install, making sure that all the dependencies were installed.

Overall, the day was filled with learning and growth as I learnt new technologies and tools. The combination of meteorJS and nvm provided me with the resources to successfully handle the project. Additionally, taking the necessary steps to set up the project allowed me to work and ensured that I had access to all the necessary components.

Thursday 30th November

Thursday was a significant day for our team because I realized the importance of maintaining consistency in the coding practices. As a result, I discovered a useful tool called prettier, which we now recommend for making code changes. Prettier ensures that the code is not only organized but also formatted in the same manner across all files.

By adhering to a specific standard for code styling, everyone in team can maintain a cohesive and professional code formatting throughout the project.

To further enhance the quality of our code, I had implemented a linter in our pipeline for every commit pushed to BitBucket. This linter checks if the code meets the stylistic requirements set by prettier. Consequently, if any code is not formatted with prettier, the project branch will fail, alerting us to the need for the styling correction.

To streamline our workflow and maintain consistency, I also incorporated prettier as part of our code review process. Before submitting a pull request, it is ensured that the code is properly formatted using prettier. By doing this, we not only ensure that the code adheres to our coding standards but also make it easier for our team to review and understand the changes being made.

Overall, the introduction of prettier has greatly improved our development process. It has provided us with a clear and defined set of guidelines for code styling, ensuring that our codebase remains organized and professional. By integrating prettier into our pipeline, we can catch any formatting issues early on, preventing them from causing major setbacks in our project. Additionally, adding or utilizing prettier into our code review process has improved the overall efficiency of our team and has made collaboration much easier.

Friday 1st December

On Friday, I found myself responsible for setting up all the necessary elements for a project. However, my day took an unexpected turn when I was assigned a ticket to fix a seemingly minor typo issue on the website. As I approached the task, I realized that it would prove to be more challenging than I initially anticipated.

As I got into the project's structure, I was faced with a disorganized mess. Navigating through the various files and directories proved to be a super difficult task, filled with frustration and confusion. Nevertheless, I remained focus to locate the exact file that required my attention.

After long and hard hours of searching, I finally discovered the file that happen to have the typo which was mentioned in my ticket and with a sense of relief, I swiftly made the necessary edits, correcting the wrongful text. Yet, my commitment to excellence did not end there. Recognizing the importance of managing to a diverse user base, I went ahead and added translations for the corrected word in two separate files: one for Finnish and another for English. This additional step ensured that users who switched to these languages would be met with accurate and appropriate translations according to the language they selected.

In conclusion, what initially seemed like a straightforward task turned into a challenging task that demanded both consistency and attention to detail. Through my passionate pursuit, I not only fixed a minor typo but also contributed to the translation assistance and user-friendliness of the website by incorporating the necessary translations in the languages.

Weekly Analysis 1

This week went well. I was extremely efficient in completing the onboarding procedure, successfully familiarizing myself with the team, policies, and setting up my development environment on my laptop. Throughout the week, I learnt new technologies like meteorJS and proficiently managing Node JS versions using NVM (Harband J s.a).

I also was able to solve an error about "semver is not a function," but by discussing with the tech lead, I engaged in a process of trial and error, leveraging online resources and team expertise to identify and resolve the issue. While the solution—deleting a node_modules folder—may have seemed unusual or unconventional, it made me realize the unpredictable nature of problem-solving in software development and the importance of wide thinking and resourcefulness.

Integrating code styling practices with Prettier (Prettier, 2017) and enforcing them within our development pipeline showcased a commitment to elevating code quality and fostering a cohesive coding standard within the team I encountered various situations that required me to utilize my strong problem-solving skills, and I even had the opportunity to learn and implement new tools. Additionally, many of the tasks I worked on demanded careful attention to small details, which I made sure to provide.

One of the most rewarding experiences of the week was the setup of my development environment in reagrds to Lahjakamu's guidelines. This involved installing the requisite Node.js version, configuring the Vite bundler, and initializing a project from scratch. I derived immense satisfaction from accomplishing them within a single day, really appreciating the effectiveness of Lahjakamu's onboarding processes and my own capacity for rapid learning and adaptation.

However, there are areas where improvement is needed. For example, I realize the importance of improving my documentation skills to ensure clear and comprehensive understanding. Moreover, I understood that better code organization is crucial for maintaining efficient and structured coding

practices. Finally, I am excited to continue my learning journey, taking advantage of opportunities to develop my skills further and gain valuable experiences.

3.2 Observation week (04-12.2024 – 10.12.2024)

Monday 4th December

On Monday, the beginning of the second week, I received a new ticket: to work on REST API tests. For such tests, I could simply utilize Postman, a popular tool for testing APIs (Postman). However, in this ticket, it was explicitly required that an external extension in Visual Studio Code (VS Code) be used, and that a new file with a .test extension be created. The purpose of this file was to serve as a testing suite that would run various test scenarios. To execute the test suite, one would first

need to input the token code obtained from the login session. After that, the individual performing the tests would have the opportunity to write the necessary code for the GET, POST, EDIT, and DELETE routes. Conveniently, right beside each of these routes, there would be an option to immediately run a test on it. This feature made it easier to verify the functionality and reliability of the API endpoints.

To enhance the functionality of the webpage, I implemented additional routes that can be accessed through their corresponding Ids. These routes provide individual unit id, allowing for specific information retrieval. The purpose of these routes was to get a specific department which matched the id. Upon completion of this task, I followed the standard procedure of creating a branch and pushing my changes to GitHub. Finally, I merged the branch with the main codebase, ensuring that all the improvements merged with the existing system.

Tuesday 5th December

Tuesday, the second day of the week, once again I was again working on backend development tasks. Today, a new ticket was assigned to me, presenting an interesting and new challenge: the implementation of custom validation rules. This particular task required a deep understanding of how user authentication and access levels function within our system. Depending on whether a user logged in as an admin, a normal user, a manager, or any other role, different limitations and access privileges would apply.

To accomplish this, I needed to make some important changes to the post, get, edit, delete routes. By incorporating an authentication token validator, I could reliably check the role of the currently logged-in user. This added a crucial layer of security, ensuring that only users with the appropriate permissions could perform certain actions. Without this validation in place, there was a risk of unintended data corruption or unauthorized access by individuals who should not have the authority to make such changes.

Once the necessary code modifications were implemented, it was essential to thoroughly test the system. I took on the role of a user with different levels of access and attempted various actions such as deleting, adding, and editing data. The validation process proved to be effective, as it successfully prevented unauthorized actions and granted access only where it was intended. This testing phase provided reassurance that the changes made were indeed functioning as expected.

With the added security measures in place, the application now had an extra safeguard against unauthorized actions. This protection was necessary in preventing any unintentional or malicious alterations that could potentially compromise the integrity of our data or grant unwanted access to sensitive information. By prioritizing the implementation of custom validation rules, I could confidently assure our users that their data and the system as a whole remained safe and protected from unauthorized manipulation.

Wednesday 6th December

On the next day, which was Wednesday, I attended my regular sprint meeting, as I do every day. During this meeting, I had the opportunity to discuss the tasks I had been working on and outline my goals for the day. It was during this meeting that I was assigned a new ticket, which addressed a login error that needed to be fixed. Specifically, the issue was that when a user attempted to log in, they were not receiving any message that their username or password was incorrect and rather

a 404 error in the console log. This lack of feedback could potentially lead to significant problems, so it was important to implement a solution.

To solve this task, I started by adding if-else statements into the codebase specifically in the user management file, using request and response as parameters from the backend. These statements allowed for different error messages to be displayed based on the specific situation encountered during the login process. For example, if the email provided by the user was not found in our system, an error message stating that the email was not registered would be displayed. Similarly, if there was a mismatch between the entered username and password, the user would be informed of this error as well. To handle all these login-related issues efficiently, a new file named "authentication-ErrorHandler" was created. This file was responsible for managing and addressing any potential login-related problems that may arise.

Furthermore, as an additional security measure, I also implemented the use of a token. This token, which is stored in the user's browser, remains active for a period of up to 2 hours. Essentially, once a user successfully logs in, their session will remain active for 2 hours after the login (considering the user becomes inactive just after logging in). However, if the user is actively using the website, their session will continue for even longer. This particular security feature is commonly utilized in sensitive areas or websites such as banks, where preventing unauthorized access is one of the top priorities.

By implementing these changes, not only I was able to solve the issue of the incorrect login error message, but we also learnt new information about the overall security of our system. This is particularly significant in industries where maintaining the confidentiality and integrity of user data is very much needed. The addition of the authentication-ErrorHandler file and the token-based session management system were necessary steps towards ensuring the login process remains secure and user-friendly. Today was another day another learning experience.

Thursday 7th December

The day started off simple, I again attended my regular daily sprint meeting and today there were no new tickets. So, I decided to make some changes throughout the code which will ensure the project is organized.

Overall, this day was dedicated to enhancing the organization and functionality of the project. From establishing a consistent file import order to rectifying the planner's authorization capability, I focused on improving the codebase's stability and usability. These changes not only contribute to a development process but also align with best practices in coding standards, ensuring that the project is scalable and maintainable in the long run.

By implementing this ordering system, I not only prevented duplicate imports but also ensured that any necessary changes or fixes could be easily applied across the entire project. With the imports consistently appearing in the same line numbers throughout the codebase, troubleshooting and making updates became much more efficient. This approach not only improved the readability and maintainability of the code but also minimized the chances of introducing errors or conflicts during development.

In addition to these changes, I also addressed an authorization issue regarding the planner functionality that I had been working on this week. During the initial implementation, I had overlooked the fact that the planner needed to have the necessary power to edit various elements. Recognizing this oversight after receiving feedback on my code review, I immediately took steps to resolve the issue. By adjusting the authorization settings, I ensured that the planner component now has the appropriate permissions to modify and update relevant information.

Friday 8th December

As this was the end of the week, there wasn't much to do, so I took the opportunity to work on improving my project. I started by attending my daily sprint meeting, where I engaged in the usual

discussions surrounding updates and goals for the day. I decided to focus on refactoring many files, an example of which can be seen with the following code snippet:

```javascript
function isEven(num) {
if (num % 2 === 0) {
return true;
} else {
return false;
}
}
```

After refactoring, the code transformed into a more concise and readable version:

```javascript
function isEven(num) {
return num % 2 === 0;
}
```

The benefits of refactoring files and functions are plenty. Firstly, it greatly improves readability. By using meaningful names and breaking down larger components into smaller, more manageable pieces, it becomes much easier for developers to understand the code. This is especially important as the project scales, ensuring that new team members can quickly understand the codebase.(Sharma I 2022)

Refactoring also enhances maintainability. As the codebase evolves and changes, having clean and organized code makes it easier to make updates and modifications without introducing bugs or compromises in functionality. This contributes to the overall stability and flexibility of the project.

Another advantage of refactoring is improved performance. By reducing redundant code and optimizing resource utilization, the code becomes more efficient and performs better. This can lead to faster execution times, better memory management, and ultimately, a more responsive and efficient application.

In conclusion, refactoring files and functions is a worthwhile as it offers many benefits. It improves code readability, enhances maintainability, and can lead to better overall performance. By taking the time to refactor, developers can ensure that their projects are built on a solid foundation that can support the growth and evolution of the application or project.

Weekly analysis 2

Throughout the week, I dedicated my time and efforts over improving the efficiency and reliability of the software development process. In a detailed manner, One aspect was that I focused on improving included the implementation of robust testing mechanisms for the rest APIS and that was done by a tool integrated within Visual Studio Code (also called VS code extension) (HuaChao Mao 2016). After setting up these test suites, it helped the development team to confidently check the quality and working of new code introduced to the project without worrying about breaking

existing functionalities. This assisted in streamlining the development workflow and also ensuring integrity and stability of the project.

I took advantage of Postman (Postman, 2012) capabilities and used the application to test the backend routes of the project or application. I used postman to check each route's endpoint and then checked the response, functionality and performance. By doing this I was able to check each route was working, so it will be a good experience for whoever will be using our application.

Moreover, I undertook the task of optimizing our codebase, cleaned its structure and readability for enhanced maintainability and scalability. Through refactoring and code cleanup, I was able to eliminate redundancies and create proper sections, making the codebase more intuitive and easier to understand and make changes to for both current and future developers. This maintenance not only facilitated better development cycles but also minimized the potential for errors.

Overall, by working and addressing various aspects of the software development process, including testing, security, and code quality, I have contributed to creating a culture of excellence and continuous improvement within our team. These efforts escalated or improved the quality and reliability of our applications or projects.

3.3 Observation week (11-12-2024 – 17.12.2024)

Monday 19th February

At the beginning of the week, I dived into my work by eagerly participating in the daily sprint meeting. During this meeting, I outlined my tasks and provided a detailed insight into my ongoing activities. It was during this session that I was handed a fresh ticket that required me to configure private routing, a task I had some familiarity with due to past experience with routing. However, I had no experience with private routing before, which prompted me to seek out additional resources for guidance.

I came across an informative blog post titled React Router 6: Private Routes by a well-known author. This blog assisted me setting up private routing effectively. I followed the guidelines outlined in the documentation provided, dedicating time and effort to understand the concept thoroughly. Through a systematic approach, I successfully implemented private routing into the project, a significant milestone in my ongoing development journey.

Following the implementation phase, I proceeded to conduct manual tests to ensure the functionality and reliability of the private routing feature. As I checked the system, I carefully examined each aspect to guarantee its integration into the existing project framework. With reassurance in the stability of the implementation, I then pushed the updated code to GitHub, creating a dedicated branch to showcase the new functionality.

As the code awaited further review , I anticipated the evaluation by our team lead, who would analyze the private routing feature for its compatibility with the project requirements. This critical evaluation served as a crucial checkpoint, ensuring that the new functionality aligned perfectly with the project's overarching goals and standards. Through this comprehensive process, I not only expanded my technical expertise but also solidified my commitment to delivering high-quality solutions within the team environment.

Tuesday 20th February

Once again the usual happened about the meeting – a routine occurrence that this time introduced something new which I hadn't encountered before. Instead of the usual tasks, a new responsibility was assigned to me: I was tasked with transferring a footer from one project to another, a task that required close attention to detail. To work on this task, I began by cloning the original project, delving into its structure to locate the specific file that held the crucial code for the footer. Upon inspecting the code, it became clear that the styling was improper or not up to the standards, prompting me to reorganize it for better coherence and maintainability.

To streamline the process and enhance the project's structure, I decided to create a separate styles file to neatly encapsulate the footer design. It was crucial to ensure that the code remained intact and functional when transplanted into the new project, a task that demanded precision and expertise to avoid any disruptions or errors. I approached the task methodically, verifying each step to guarantee a transition.

Given that the Create React Application (CRA) framework was no longer receiving updates, I made the strategic decision to use Vite for the new project. Vite's advanced features and exceptional speed, being twenty times faster than traditional CRA, offered a more efficient development environment. Leveraging Vite, I swiftly set up the new project, organizing the project structure to

accommodate the footer file. Placing the footer file within its designated folder, I ensured that the project's architecture was optimized for clarity and future scalability.

Wednesday 21st February

Tomorrow, once again, I did the routine of attending the daily sprint meeting. However, this time around, my focus shifted towards working on the header component. Recalling the tasks from the previous day, I recalled the procedure for the task which I did yesterday and had undertaken when dealing with a different aspect of the project. With focused attention to detail, I carefully extracted all the necessary and vital code snippets relevant to the header section. Ensuring a clean separation from any code dependencies in other files, I minimized the likelihood of encountering errors in the future, thereby creating or assisting in a better development process which would help the project.

Subsequently, I integrated the newly crafted header component, developed using React, into the existing VITE project that had been initialized the day before. Eager to validate the functionality and visual appeal of the header, I conducted a series of tests, checking its performance across various screen resolutions and ensuring its responsiveness on different devices. As I reached the end of this task, I prepared to seek feedback and approval from my peers by initiating a pull request and committing the changes to the project repository.

As time progressed, the efforts resulted into refining the header component were acknowledged and endorsed by the team lead, supporting the successful completion of the task.

Thursday 22nd February

Today, my workday was free from any scheduled meetings, allowing me to focus on a crucial aspect of my project: the integration of the i18next translation module. This module posed a necessary learning curve for me to ensure the implementation of translations across all essential routing pages. Beginning my journey into this new territory, I embarked on a comprehensive exploration of the official documentation available at the official website of i18 (Getting started with i18-next)

Taking a proactive approach, I initiated the installation of the i18next library and proceeded to set up a dedicated configuration file, named i18config.js. Within this file, I assembled all the essential languages required for the translation process. The structured setup helped me lay a solid foundation for my future integration efforts. Researching deeper into the structure or system of the module, I familiarized myself with the specific format and guidelines outlined in the documentation. This preparatory step was instrumental in equipping me with the necessary knowledge to navigate the translation process efficiently.

My next crucial task involved the systematic incorporation of translations into all public routing pages across the website. Through implementation efforts, I ensured that every page incorporated the multilingual support provided by the i18next module. Completing this comprehensive localization, I reviewed and verified the changes made before initiating the final step of pushing my modifications. Following the industry-standard protocol, I thoroughly tested the implemented translations

to validate their accuracy and functionality. Only after rigorous testing and validation did I confidently submit my pull request for review and approval, ensuring that the implementation met all specified requirements and standards.

Friday 23rd February

After the usual daily sprint meeting, where I provided updates on my progress, discussed issues, and received new tasks, I was assigned a new ticket. This ticket required me to develop the frontend for a redeem giftcard application. Utilizing axios, I successfully established the connection between the frontend and backend systems. Once the setup was complete, I immersed myself in crafting the frontend elements. I designed the redeem page to display crucial information such as the remaining balance and the company's name, ensuring a user experience. To maintain consistency with the website's design pattern, I styled the page in colors that matched with the brand identity.

Moreover, I extended the application's functionality by implementing two additional routes. The first route assisted or went into the redemption process, guiding users through the steps required to reclaim their gift card value. The second route led users to a thank you page, which would be displayed once the code was successfully redeemed. As the application was in the testing phase, I concentrated on fetching and displaying a single ID from the backend to simulate real-world scenarios. Notably, the thank you page also featured the transaction ID, providing customers with a reference point for future interactions. Overall, this project not only showcased my technical skills but also highlighted my attention to detail in creating a user-centric interface for the gift card redemption process.

Weekly analysis 3

This week I focused on enhancing our project's functionality and user experience by going into a series of technical tasks, with each task having an important role in achieving the development objectives.

Beginning of the week I started with the implementation of private routing using React Router 6, I explored into a well-crafted blog post (Wieruch R (February 6, 2022)) to gain valuable knowledge and with a clear understanding of the guidance provided, I navigated through the process of setting up private routes, guaranteeing that only authenticated users could access sensitive content. This strengthened the security of our application but also improved user interactions by assisting them towards content that should be allowed based on their authentication status. Through testing and validation, I successfully confirmed the integration of private routing, thus securing our project against unauthorized access and potential security risks.

Moving on to the migration of footer and header components across projects, I used the efficiency of Vite's build tool. There was no point in using Create React Application (CRA) framework due to its limitation of being too slow, I made a strategic decision to adopt Vite to improve development and enhance project maintainability. By thoughtfully organizing project structures and optimizing component architectures, I assisted in the integration of footer and header components into the new project setup. This not only ensured consistency in design and functionality across projects but also made it easy for future scalability and extensibility.

Then I integrated i18next for language translation across various pages marked a significant use in the application towards internationalization and localization (i18n). I was able to make proper use of the official documentation of i18next (Mühlemann J i18next (2011)), I guided the setup and configuration of the i18next library to enable multi-language support within our application. By focusing on the detail and exhaustive testing, I ensured the accuracy and functionality of language-switching capabilities across different pages. This extensive localization effort not only improved user accessibility but also made the project for broader global visibility and acceptance.

Throughout these tasks, I prioritized continuous improvement in testing methods, learning strategies, and team collaboration to drive project success. Moving forward, I'm committed to maintaining this focus on growth and innovation.

3.4 Observation week (26-02-2024 – 03-03-2024)

Monday 26th February

At the start of the new week, I researched into a fresh set of tasks or topics, diving straight into my regular daily sprint meeting where I updated my team lead on the progress and completion of the tasks I had been working on since last Friday. This week holds exciting new tasks that I aim to collaborate closely with the team lead on handling the new tasks and tickets, particularly focusing on projects involving Amazon Web Services (AWS). One task or challenge that I received involved the establishment of an AWS S3 service, which is known for its simple yet secure storage solutions. The significance of this task prompted me to familiarize myself with the complication or issues of the AWS S3 service by properly studying the official documentation (Amazon 2024b). With a solid grasp of the service's functionalities, I proceeded to follow a comprehensive guide that walked me through the process of creating buckets within the system and setting up repositories using CodeCommit, another integral AWS service.

Dedicating myself to this task, I navigated through the AWS console, using PowerShell to clone, create backup repositories, and add content. The process involved a strategic approach as I ensured that the new repositories were not only created error free but also populated with some sample content before effectively pushing the changes to my planned repository. The task demanded a high level of attention to detail, as any misstep could potentially impact the integrity of the system. Through the methods of steps and focus, I successfully completed the initial phases of establishing

the AWS S3 service, paving the way for further enhancements and optimizations in our cloud infrastructure.

As this week unfolds, I am excited to continue my collaboration with the team lead and go deeper into AWS, exploring new functionalities, services, and further expanding my skill set to contribute meaningfully to the team's ongoing projects and initiatives. Each task presents a unique challenge and an opportunity for growth, and I am committed to navigating through them with diligence and enthusiasm, ensuring that each milestone is achieved with excellence.

Tuesday 27th February

In today's meeting, I engaged in an in-depth discussion regarding my progress with AWS S3, elaborating on the intricacies of the project and the milestones achieved this far. The team acknowledged my efforts and trusted me with a crucial task: the deployment of a static website utilizing AWS S3 services. Me being excited to start this new task, I launched the AWS Management Console to initiate the necessary setup process.

With a sense of determination, I proceeded to install the AWS Command Line Interface (CLI) on my system to facilitate a interaction with AWS services. Once the installation was successfully completed, I used Git Bash, a command line terminal, to begin configuring the AWS CLI. I initiated the configuration process by entering the command "configure," prompting the system to request my access key ID and secret access key for authentication.

Carefully entering the required credentials, I navigated through subsequent prompts that sought additional information, such as the desired output format and server location. Mindful of the project

specifications, I planned the options, ensuring a streamlined setup tailored to the site's intended server environment. With focused attention to detail, I completed each configuration step, ultimately achieving a fully configured AWS CLI for efficient website deployment.

Wednesday 28th February

After attending the typical daily meeting as usual, I quickly shifted my focus back to the task I had been working on the day before, recognizing the need to further develop it in order to have it up and running live on AWS. Upon accessing my bucket on AWS, I meticulously adjusted the necessary permissions, ensuring that every detail was in place.

I proceeded to transfer the content, previously stored locally, directly to the bucket using the efficient aws sync command, all the while prioritizing the essential step of making the content publicly accessible, a crucial requirement for impending user interaction. By going into the settings, I disabled the pre-existing "Block public access" checkbox. Turning my attention to the properties tab of the bucket, I activated the static website hosting feature, a pivotal step towards the goal of establishing a live website environment.

After all this came the entering the index document's name into the designated field, I saved each crucial change, marking significant progress towards the functional realization of the website. With every step completed successfully, the website transitioned into a live and fully operational state, operating flawlessly without any 403 errors hindering user accessibility.

Thursday 29th February

After my meetings, the following day I explored further into the diverse range of services offered by Amazon Web Services (AWS), specifically into AWS Lambda, a powerful compute service that changes the way users execute code and manage back-end services. Essentially, Lambda liberates users from the hard part of server management by providing a platform where code can be effortlessly written directly on the console and executed on the robust AWS servers. This not only streamlines the operational process but also empowers users to focus more on code logic and functionality rather than the infrastructure.

The exceptional advantages of AWS Lambda extend beyond its ease of use; the service is tailored for elasticity and scalability, dynamically adjusting its capacity in response to varying workloads. This automatic scaling feature ensures that resources are efficiently utilized, paving the way for a hassle-free and resource-efficient operation. (Amazon 2024c) Furthermore, Lambda's "Pay-as-you-go" pricing model emerges as a compelling solution for enterprises of all sizes, eliminating the need for upfront investments in infrastructure while offering a cost-effective billing system based on actual compute usage.

My journey with AWS Lambda led me to create a custom Lambda function, making my understanding better of the service's capabilities. Navigating through the console, I initiated the creation process by locating the Lambda service and creating a new function with a simple right-click and selection method. By defining the desired function type and selecting the appropriate runtime environment such as Node.js, .NET, or Python, users can tailor their functions to meet specific requirements. This versatility in runtime languages amplifies Lambda's appeal, offering a wide array of options to serve or handle the diverse developer preferences and project needs.

Upon finalizing the function creation process, users are greeted with an clean interface, showing the code source environment and an interactive console display. This visual representation not only reinforces the user-friendly nature of Lambda but also displays the integration between code development and execution, enhancing the overall user experience. In essence, AWS Lambda emerges as a work changer in serverless computing, making the way for efficient, scalable, and cost-effective cloud solutions for businesses and developers.

Friday 1st March

The next day, after returning, I went back into the task which I was currently managing. I diligently proceeded by crafting the necessary code within the index.mjs file. To ensure the code's functionality, I tested it by populating the Event JSON with the expected outcomes. Furthermore, I dedicated time to assigning a name to the test and securely saving the configuration settings.

Upon completing the code preparation, I initiated the test execution by simply clicking on the "Test" button. This action triggered Lambda to effortlessly run the test, generating results that were promptly displayed within the code source pane for my review. As part of the comprehensive testing process, users have the option to make use of Log groups, an invaluable tool that allows for deeper monitoring and analysis of the code's execution. By leveraging Log groups, users can guarantee the efficiency and accuracy of their Lambda functions, ensuring optimal performance.

With all the essential testing and verification steps completed, I moved on to the crucial task of generating a shareable URL that would grant access to my team members. By making the code accessible to everyone on the team, we could collectively ensure that the code underwent thorough testing and validation. This collaborative approach helps in identifying any potential issues or areas for improvement, fostering a culture of teamwork and excellence within the team's development process.

Weekly analysis 4

Throughout the week, I focused on several key achievements in my professional development. This included configuring AWS S3 services (Amazon 2024b) in detail and successfully deploying a live static website using AWS CLI. Additionally, I delved into learning AWS Lambda (Amazon 2024c), deepening my understanding of serverless computing through developing custom functions and rigorous testing for optimal performance.

This week I had started with the intricate task of setting up an AWS S3 service, powerful cloud storage solutions renowned for its user-friendly simplicity and robust security measures. Along with my strong understanding of AWS S3's functionalities gained through studying of the official documentation; I navigated the AWS console with a keen focus on precision. Using PowerShell for streamlined repository management and content distribution, I ensured a process of creating buckets and repositories, laying a sturdy groundwork for future enhancements and optimizations in our cloud infrastructure.

I was also tasked with the deployment of a static website using AWS S3 services, a project that used the practicality of cloud computing in modern web development. I configured the AWS Command Line Interface (CLI), enabling effortless interactions with various AWS services. Initiating the configuration through Git Bash, I prioritized the project specifications and designed optimal server environment settings. This approach resulted in the successful deployment of a fully operational static website, showcasing my speed to adapt in utilizing AWS tools for tangible real-world applications.

As I worked more on AWS, focusing on improving the deployed website by fine-tuning permissions and ensuring widespread public accessibility. By using the efficient 'aws sync' command, I transferred content to the bucket while ensuring key steps like enabling public access and activating static website hosting were executed promptly. This iterative process led to the website transitioning into a live state, operating without accessibility issues, and created user engagement.

While assessing my progress, I identified areas for improvement to enhance workflow and team dynamics. Strengthening documentation practices, refining task prioritization strategies, and fostering better communication and collaboration within the team emerged as crucial objectives for future growth and efficiency.

Looking ahead, I'm dedicated to maintaining a continuous learning mindset and prioritizing these strategic objectives. By doing so, I aim to not only advance my personal development but also significantly contribute to the overall effectiveness of the team in achieving our cloud computing goals. Through ongoing improvement and a culture of learning, I am confident in our ability to overcome challenges, seize opportunities, and make meaningful progress together.

3.5 Observation week (04-03-2024 – 10-03-2024)

Monday 4th March

Today, I had my normal meeting where I discussed about my updates and got assigned a task to implement function with lambda and use terraform to implement it. I studied and implemented lambda function with help of terraform. What is terraform? Terraform is a tool used for Infrastructure as Code (IaC), which basically helps you to manage your infrastructure resources like servers, databases and networks or settings associated with them by using a declarative configuration file which ends by .tf file extension.

Some of the benefits I explored while working with it were
Declarative configuration: As it enables you to specify the desired end state of your infrastructure. For example if you wish to have a state of AWS S3 bucket instance you can simply mention it as -> resource "aws_s3_bucket" "example_bucket" {other props}

Multi-Cloud support: If you happen to switch from AWS to Azue you can easily configure the .tf file (provider.tf) to provider "azrurerm" and not having to configure all the things again as that can be time consuming.

Automated Resource Lifecycle Management: Whenever you update a resouce/file, you can simply run "terraform apply", and it will automatically find the necessary changes and applies them,

ensuring infrastructure matches desired state. It handles the entire lifecyle of resources, including creation, updates, and deletions, in such a way that it minimizes disruption.

That’s all what I got to learn and work on today.

Tuesday 5th March

After today’s meeting I was assigned another task with AWS, which was to implement another AWS service called CloudFront (Amazon 2024a), which plays a crucial role in efficiently delivering content to users across the globe. CloudFront essentially functions as a content delivery network (CDN) that leverages a widespread network of data centers known as edge locations.

To understand this concept further, consider the scenario where you're streaming a YouTube video. The reason for the loading experience is attributed to the fact that the video content is cached closer to your geographical location. As a result, the video doesn't have to travel long distances from its original storage location, which could be several thousand kilometers away, thereby preventing latency issues that could impede the video playback speed. By utilizing AWS CloudFront, content is delivered based on the principle of minimizing latency, ensuring optimal performance and user experience. This service offers plenty of advantages which included:

It proves to be a cost-effective solution by diminishing bandwidth costs since the origin server is relieved from the burden of constantly fulfilling data requests. Consequently, the volume of data transmitted from the origin server is lowered. Additionally, CloudFront enhances content

availability by efficiently managing web traffic, alleviating server loads, and providing operational redundancies to ensure uninterrupted 24/7 service delivery.

After studying about it and reading the documentation was time to start implementing it step by step.

Wednesday 6th March

At the beginning of the day, I resumed working on what I had learned the day before. My focus was on the practical implementation of using AWS CloudFront to enhance the performance of a static website. To achieve this, I started the process by setting up a basic S3 bucket. This involved transferring data from my local repository, which was on my system, into the S3 bucket using the Amazon CLI. Alternatively, to streamline the setup, users have the option to configure the AWS CLI or employ a more user-friendly approach of dragging and dropping files and folders directly into the designated bucket.

Once the data transfer was completed, the next step was to access CloudFront. I navigated to the CloudFront service by using the search bar and proceeded to create a new distribution. During this step, I assigned a suitable name to the distribution, ensuring that it accurately reflected the origin domain name. It's essential to note that when setting up the S3 bucket, attention should be paid to ensuring that the naming convention aligns with the required format, such as ending with .com or .co.

After defining the origin name and finalizing the distribution settings, I confirmed the setup by clicking on the 'create distribution' button. With the distribution now in place, I located the specific distribution domain name within the Distributions section. This unique link provided direct access to the

website that was now being served securely through CloudFront, optimizing its performance and accessibility. By following these steps meticulously, I successfully established an efficient cloud-based infrastructure for hosting and delivering a static website.

Thursday 7th March

Today, when the scheduled meeting was unexpectedly cancelled, I started finding task by myself, seeking out and managing tasks independently. To tackle the situation, I explored into the project details, discovering numerous variables that were not in line with the established restful naming conventions. Taking a methodical approach, I dedicated myself to correcting inconsistencies such as standardizing names like retrieveGood and getId to align more closely with the desired structure, specifically updating all get requests to commence with getProduct and getId. This process extended across various request types, encompassing get, post, put, and delete, amidst which I rectified through each file to ensure all discrepancies were resolved.

In addition to correcting the naming discrepancies, I checked through the codebase to eliminate any code of the outdated variable names, recognizing the potential complications they could pose during the application's execution. Upon completing the renaming process, I tested the project to validate the functionality of all components. Finally, I crafted a comprehensive pull request, outlining the purpose of my modifications in detail, which got approval before eventually being merged into the main project repository following a brief period for review and validation.

Friday 8th March

Today, I got assigned a new task where I had to work on password hashing. I started off this task with Bcrypt. Bcrypt is a hashing function primarily used for securely storing passwords. Bcrypt operates in a way that it takes a password and salt (random data) as inputs and produces a fixed-size hash value as output which is used when a user logs in to a website and puts in the password, the password is then converted to hash and checked with the already stored hash value for authentication.

In order to use this, I simply installed it by "npm install bcrypt", then imported it into the file by const bcrypt = require('bcrypt')

Then a function was written in this way (sample code)

```
const bcrypt = require('bcrypt');

async function generateSalt() {
try {
// Number of rounds for generating the salt, optional*
const saltRounds = 10;

// Generate a random salt using function .genSalt
const salt = await bcrypt.genSalt(saltRounds);
```

```
console.log('Generated salt:', salt);
} catch (error) {
console.error('Error generating salt:', error);
}
}

generateSalt();
```

The benefits of using Bcrypt for this task were:
Makes guessing passwords super hard: Like adding a ton of locks (cost factor) to your vault, making it take forever for someone to crack it (brute-force attack).

Salts your passwords for uniqueness: Bcrypt adds salt to each password, making them all unique and impossible to guess from pre-made lists.

Hides your passwords completely: Bcrypt doesn't store your passwords directly, but a secret code (hash) instead. Even if someone finds the codes, they can't crack them back into your passwords.

Popular and works well with many things: Bcrypt is like a widely used lock that works with many different doors (programming languages).

Weekly Analysis 5

This week I learnt and explored more about cloud computing technologies, along with hands-on implementation of various AWS services and tools. My focus was on technical tasks involving AWS Lambda functions, terraform (HashiCorp (2014)) for infrastructure management, AWS CloudFront for content delivery, and password hashing with Bcrypt (Npm,2010) has significantly enhanced my skill set and contributed to my professional growth.

I explored terraform which is a robust solution for implementing Infrastructure as Code (IaC), with its efficiency it brings to overseeing infrastructure components through the utilization of declarative configuration files. This hands-on experience not only deepened my understanding of Terraform but also emphasized its advantages such as the ability to define infrastructure states declaratively,

the convenience of transitioning across multiple cloud platforms, and the automation of resource lifecycle management to enhance the operational upkeep of infrastructure.

Additionally, I worked with implementation of AWS CloudFront, a content delivery network (CDN) service designed to optimize content delivery and enhance user experience. By leveraging CloudFront's widespread network of edge locations, I gained a deeper understanding of its advantages, including cost-effectiveness, enhanced content availability, and efficient web traffic management. Through meticulous setup and configuration, I successfully deployed CloudFront to serve a static website, optimizing its performance and accessibility.

I fixed the naming inconsistencies and outdated variables within our project, demonstrating my commitment to code quality and adherence to established naming conventions. Through methodical renaming and code cleanup, I ensured the functionality and excellence within the team's development process.

I worked with a task that involved password hashing with Bcrypt, a secure hashing function used for storing passwords. By implementing Bcrypt and understanding its benefits, including enhanced password security, unique salt generation for password uniqueness, and complete password protection, I strengthened data security measures within our project.

3.6 Observation week (11-03-2024 – 17-03-2024)

Monday 11th March 2024

Today was the beginning of a new week, and as usual routine, I began my day with the usual daily spring meeting. During this meeting, I was given a new task that required me to utilize the i18next feature developed earlier for a new project. The purpose of this task was to ensure a translation of languages without encountering any errors. To start with the process, I began by transferring files from the previous project and meticulously reviewing the code.

Unfortunately, this initial phase was filled with numerous errors. However, through the installation of the necessary library and the creation of additional files, most of these errors were successfully resolved, though a few were still present.

As the day progressed, I found myself running out of time, leaving me with no option but to continue the work on the following day, aiming to fix remaining errors and finish the task.

Tuesday 12th March 2024

After having the meeting with my team to discuss the updates on my tasks, I proceeded to work on fixing the language translation feature for the new project I was working on. The issue I encountered was a technical error in my code that was preventing the language from switching when a different language was selected. Upon closer inspection, I identified that the problem came from just one line in my code, which took some time to be found. It turned out that this particular line was causing the language to default to Finnish every time the page was refreshed, overriding the user's language selection. This was quite a significant oversight on my part, leading to a situation where the language preference was not being retained as planned by the function.

To fix this, I needed to revise the code to ensure that the language selection was accurately stored in the browser, allowing it to persist even when the page was refreshed or revisited. So, part of my task involved not only fixing the initial coding error but also implementing a solution that would maintain the selected language consistently throughout the user's browsing experience. This process of debugging and enhancing the language translation feature was crucial for achieving a and user-friendly interface for our project.

Wednesday 13th March 2024

Today there were no meetings scheduled, which provided me with the perfect opportunity to explore into the project independently. I took the initiative to thoroughly explore and examine different components, focusing particularly on testing the frontend. During this testing phase, I encountered an issue with the hamburger menu disappearing when the screen size was adjusted to a smaller view. In order to troubleshoot this problem, I decided to conduct tests using the normal responsive menu settings within the browser.

After identifying the root cause of the problem, I implemented a solution by integrating a media query that would ensure the hamburger menu remained visible once the screen size reached a specific limit. This adjustment was crucial in enhancing the user experience across various devices, including smaller laptops, larger screens, and standard smartphones. Once I had successfully implemented these changes and thoroughly tested them, I proceeded to create a pull request that contained all of my modifications and improvements. Subsequently, I submitted the pull request for review and approval by the relevant stakeholders, thereby ensuring that the necessary changes were acknowledged and integrated into the project.

Thursday 14th March 2024

Today, after finishing of the usual daily sprint meeting, I received a new project assignment. This time, the code had been composed by another developer and unfortunately, it was quite disorganized, posing potential issues for scalability down the road. The primary concern revolved around the utilization of in-line CSS for styling the page components, a practice that lacked efficiency and could lead to complications as the project grew.

Assigned with the task of fix this problem, I started on by extracting these styles from the main file and relocating them to a designated CSS file for better organization and ease of use. Despite the considerable amount of time and effort this task demanded, I diligently worked on converting the entire code base, ensuring that the transition maintained all styling features such as colour schemes, flex properties, and others. Following thorough testing to guarantee that all visual elements remained as they were working initially, I proceeded to send a pull request to merge my changes into the main project repository.

Friday 15th March 2024

Today, as per the usual routine, I attended the regular meeting where we discussed various updates and tasks that have been working on. During this particular meeting, I was designated the task of resolving an issue related to a backend API route. This specific route was responsible for retrieving crucial user data from the database. The presence of an error in this route was blocking the transfer of data from the backend to the frontend, thus depriving users of valuable information. Recognizing the potential impact of this malfunction on our customers, it was quickly identified as a priority for immediate fix.

Checking into the backend route issue, I spent a significant amount of time, approximately an hour, to carefully debug the code. Through careful examination and troubleshooting, I successfully pinpointed the root cause of the problem. By refining and optimizing my function for data retrieval, I was able to fix the issue and establish the flow of data transmission from the backend to the frontend. Using tools such as axios, I efficiently retrieved the necessary data and integrated it into the frontend of our website, ensuring that users could access and interact with the information and error free.

Weekly analysis 6

Throughout this week, I worked on technologies related to integrating language translation functionalities through i18next (Mühlemann J 2011) to the process of restructuring disordered codebases and tackling backend API glitches, my skills and insights have improved significantly as I worked and solved these issues.

I started off by working on the task of incorporating language translation features using i18next on It demanded my attention to detail as I looked through the code, tackling debugging challenges head-on to ensure flawless language switching functionality. Despite facing initial obstacles, being persistent led me to overcome most hurdles by integrating essential libraries and crafting additional files. This hands-on experience not only added depth to my comprehension of multilingual support

implementation but also reinforced the immense value of thorough testing procedures.

My focus later shifted towards fixing the technical glitches linked to the language translation feature. Identifying and rectifying a critical oversight in the code ensured that the selected language persisted throughout the user's browsing journey. This revision process demanded not only pinpointing and rectifying coding errors but also making sure that upon refresh the page doesn’t forget the selected translation language on the webpage. Through code alterations and optimizations, I successfully engineered a user-centric language translation module for our project.

Then I worked towards independent discovery and resolution of frontend component issues. Focused on enhancing user experience, fixed problems like the disappearance of the hamburger menu on smaller screens. Making use of a responsive design approach involving media queries, I skilfully improved the interface's adaptability to diverse devices, showcasing my adeptness in design principles.

I had to make sure that this project would be scalable in future and for that I started refactoring the code by extracting in-line CSS styles into a dedicated CSS file, executing a transition process while conserving stylistic elements. Rigorous validation and testing safeguards guaranteed the preservation of all visual components.

A critical backend API was creating issues and errors in essential data flow from backend to frontend. Immersed in debugging and refining the data retrieval mechanism, I successfully remedied the setback, reinstating the transfer of vital user data. This process of debugging and being patient with code as well as tracking its flow assisted my skills.

The technical challenges encountered this week has been very knowledgeful in my professional development, emphasizing the significance of code, debugging, structural orderliness, and scalability in software development. I am skilful in my commitment to applying these refined skills and new experiences to work with new projects or applications.

3.7 Observation week (18-03-2024 – 24-03-2024)

Monday 18th March 2024

Today was the beginning of the week, and as always, I started my day with the routine sprint meeting where I received the assignment to improve the gift card display. The specific task was regarding aligning the style of the display according to a figma design that had been provided to me. Upon going into the assignment, I encountered an obstacle in the form of the swiper from MUI, which was causing problems by limiting the flexibility of the gift card component. To address this

issue, I decided to create a customized component that would allow me to exert greater control over the CSS properties, particularly to ensure the even spacing of the cards on the display.

In addition to this technical issue, I also focused on improving text elements of the display. This involved working on the heading and description to ensure they were formatted correctly. I made sure that the title stood out by making it bold, while the description was in a lighter font to provide a visually appealing contrast. I conducted thorough testing to verify the effectiveness of the implemented changes and once I was satisfied with the outcomes, I initiated a pull request to merge my modifications into the existing branch and merging my changes.

Tuesday 19th March 2024

Once again, I followed my routine of participating in the daily sprint meeting where a new task was assigned to me. This task involved the implementation of eslint in the upcoming project, following a predetermined structure. To start the process, I initiated by installing the eslint library through the npm command 'npm install eslint'. With the library successfully added, the next step involved creating the eslintrc.cjs file and customizing the format according to the project requirements.

Configuring details such as line break style, quotes, and handling trailing spaces were crucial steps in setting up eslint appropriately. Upon completing the configurations, I saved the file and proceeded to execute the linting process by running the command 'npm run lint'. This action highlighted areas within the codebase that needed correction to comply with eslint guidelines. I fixed all identified errors to ensure the code was upto to the specified standards. Once the linting process was successfully completed, I submitted a pull request to merge my changes into the project codebase. The implementation of eslint not only enhanced the overall organization of the project code but also facilitated improved code readability and maintainability.

Wednesday 20th March 2024

Today, once again, during the regular meeting, I was provided with a new task regarding enhancing the frontend testing aspect of the project. The specific objective was to implement a system that assigns specified variable types in the frontend to prevent incorrect data from being entered into the database, which has the potential to lead to critical errors. This assignment involved using a library called prop types (Ahtari A. 2023), which enables the categorization of variables as integer, string, etc., thus ensuring data integrity. By implementing prop types, we also guarantee that the required values are indeed being passed, preventing any instance of missing or empty data.

Completing this task and conducting thorough testing, I proceeded to submit a pull request to integrate my changes into the project. In due course, the team lead reviewed and approved the pull request swiftly, recognizing the benefits that the prop types feature brings to our codebase. The utilization of prop types is crucial in preventing errors and establishing a robust system for handling variables effectively.

Thursday 21st March 2024

Today, during my routine daily meeting, I was assigned the important task of incorporating Cypress tests in the frontend to enhance the efficiency of our workflow process. To effectively carry out this task, I started studying into various resources such as documentation and instructional videos relating to interacting with Cypress. Subsequently, I installed Cypress, ensuring ESlint was enabled, and proceeded to organize my workspace by creating dedicated folders for both the backend and frontend components. I made a crucial modification to the package.json file for the backend portion, aiming to implement a specific test version.

Transitioning, I used VS Code's capability to work with multiple projects simultaneously, utilizing a split terminal to manage all three projects conveniently. Following this setup phase, I managed with creating a preliminary test scenario to verify the presence of essential elements like the username field on the login page – a test that gave positive results, validating its successful execution.

Moving forward, the subsequent tasks on my agenda involved adding tests to comprehensively evaluate the frontend features and overall user experience. While progress was substantial, there remained more testing procedures waiting for completion.

.

Friday 22nd March 2024

Today, I didn't have any scheduled meetings, allowing me to fully dedicate my time to the ongoing task related to cypress that I had been working on since yesterday. Going back into it, I began the day by creating a variety of tests to explore the capabilities of the tool. I was pleasantly surprised by the and efficient process; my workflow transitioned from defining tests within describe blocks to setting up the necessary preconditions before each test execution.

This involved navigating to the website, entering test credentials, and initiating the login process. Once successfully logged in, I proceeded to create tests that validated user actions such as deletion, addition, and liking of content. The day was incredibly enlightening as I continued to enhance my understanding of the tool's functionality. As each test started to successfully pass individual validations, I felt a sense of accomplishment. To ensure that my progress was combined with the team, I submitted a pull request to update the project on GitHub,

Weekly analysis 7

Throughout this week, my primary focus was around improving various aspects of the project, ensuring enhancements in the frontend user interface and the integration of linting and testing frameworks.

When exploring into the aspect of improving the display, I faced challenges particularly with the swiper functionality but by creating a customized component, I gained better control over CSS properties, ensuring consistent spacing among gift cards and aligning with design specifications provided in a figma design. Additionally, attention to text element formatting improved visual appeal and readability, enhancing the overall user experience.

Furthermore, attention to the formatting of text elements significantly elevated the overall visual appeal and readability of the interface. The integration of eslint (The Open JS foundation, 2013) played an important role in the project's code quality standards, requiring detailed configuration to align with specific project needs. Customizing eslint settings, such as line break style and handling trailing spaces, ensured code consistency and adherence to coding standards. The linting process facilitated improved code readability and maintainability, contributing to overall code quality.

The implementation of prop types (Ahtari A. 2023) to enforce variable type validation in the frontend, thereby enhancing data integrity and preventing errors. By categorizing variables and ensuring the presence of required values, prop types established a robust system for handling variables effectively. Thorough testing and validation of prop types of integration ensured integration into the project.

Exploring Cypress (Cypress.io (2017)) testing provided great insights into end-to-end testing, showcasing its efficiency in validating user pathways and guaranteeing the functionality of the application. Reflecting on the tasks done this week, it becomes clear that the learning of new technologies and frameworks, complemented by comprehensive documentation and collaborative efforts, assists not only the projects but my skillset massively as well.

In conclusion, this week's technical tasks have significantly enhanced project functionality, code quality, and user experience. The integration of linting and testing frameworks, along with attention to frontend design and functionality, has led the project to improve in plenty of ways.

3.8 Observation week (25-03-2024 – 31-03-2024)

Monday 25th March 2024

Once again, I made my way to my office, prepared to attend in the daily sprint meeting routine. During this meeting, I provided updates on my progress and was given a task that involved

addressing issues with the frontend design relating to the gift card layout. My focus to tackling this task was assigned with a ticket right away, I made sure to replicate the designated template accurately. Drawing inspiration and guidance from the figma design, I encountered challenges with the styles not aligning perfectly as intended.

Finding out the discrepancies step by step, I started making subtle adjustments while also seeking the precise components that required modification. As my efforts progressed, I was about to complet the ticket; however, spacing issues were still present that came to my attention. Despite my attempts to rectify these concerns, the desired results were not immediately correct. On the good side, other style adjustments, such as padding, highlighting areas, and implementing hover effects upon cursor interaction, were successfully implemented.

In refining the design further, I opted to streamline the gift card item presentation by removing the item description, retaining only the essential title. While progress was significant, a few remaining details still demanded my attention and refinement, tasks that I have scheduled to address promptly during the upcoming workday.

Tuesday 26th March 2024

Today, I returned to address the issue with the frontend giftcard design ticket. My main focus was on enhancing specific styling properties within the giftcard array. The challenge I currently encounter is the excessive spacing between each giftcard in the array. Moreover, I need to ensure there is adequate spacing on the right and left sides, and the giftcard array is properly centered. It's crucial for me to correct these aspects.

To resolve the issue, I have to carefully examine the component tree and make the necessary adjustments. I began investigating the areas that required fixing. I implemented numerous changes within the giftcard component itself and the container tree and the giftcards components. This involved modifying the gap, padding, margin, justify-content, border radius, and several other elements. After implementing all these alterations, the presentation significantly improved, and the layout looked more aesthetically pleasing. Additionally, I had to switch from using a display grid to a display flex approach in order to achieve the desired layout. I used border radius to exactly determine what component was where, which made it a lot easier for me to understand which component was under what.

By fixing these issues and refining the styling elements of the giftcard array, I was able to enhance the overall design and functionality of the frontend. Then I finally made a pull request and pushed my changes to the branch.

Wednesday 27th March 2024

Today once again, I had my normal daily sprint meeting where I engaged in discussing my recent updates and progress on ongoing tasks. During this meeting, a new task was assigned to me, which involved implementing a searching filter with Redux. This task required me to create multiple files - specifically, 2-3 files were necessary for this task. These files included essential components such as the reducer, action creators, and a combined reducer for the store.

In creating the new reducer, I began by establishing a constant variable and defining a function called setFilter. This function contained two key properties: a type property set to SET_FILTER and a payload property representing the value of the filter itself. Subsequently, I developed the filterReducer function, responsible for managing the state of the filter within the application. This function accepted two parameters, state and action, and was then exported using the default export syntax as filterReducer.

Moreover, I worked on a separate file called filter, which was used for displaying an input field for user interaction. Through this input field, users could input values that were processed using the setFilter function from the reducer class. While there were additional tasks and components that needed to be addressed in this project but that was the work for tomorrow.

Thursday 28st March 2024

Today again I had my normal daily sprint meeting where I discussed the updates from yesterday on the redux filter method. I shared what has been done so far and what was my goal for today. I started creating another reducer but for the content or giftcards. For that I had to create a reducer function once again and implement almost the same things or logic as the yesterday one.

This had the things such as different action types such as different cases: like, favourites, etc. and a specific return logic for each case. After my reducer was created, next task was to create a combined reducer which would combine my filter reducer as well as the giftcard reducer. I created separate file called combined reducer and used a function from redux library called combineReducers

which was given 2 slices or parameters of the state: giftcard and filter. Then this was exported as export default rootReducer.

After this was done, I had to make 2 minor changes, in my App.jsx where I executed the Filter component, and other change was in main.jsx where I used const store = createStore(rootReducer) and that create store function coming from the redux library. Then I finally pushed my changes to github and waited for the review.

Friday 29nd March 2024

Today, I attended my usual daily sprint meeting where I provided updates on my progress and outlined my goals for the day. One of my main objectives was to enhance the frontend by implementing Redux state management across additional files. For data retrieval from our Mongo database (Tarud J. 2024), I utilized the Axios library, while simultaneously working on a feature to generate and transmit gift cards to the backend through post requests. Within these specific files, I incorporated the React useState hook to manage the component's local state effectively.

To fully integrate Redux into these functionalities, I initiated the creation of a dedicated Redux folder consisting of three distinct types of files: actions, reducers, and the store. The actions file contained event triggers responsive to user inputs, which in turn facilitated updates in the application's state. Moving on to the reducers file, I defined functions that accepted the current state and an action, subsequently returning an updated state based on the interactions.

Finally, in the store file, I instantiated a Redux store using the 'createStore' function from the Redux library. This function required a reducer as a parameter, serving as the key component responsible for managing and updating the application's global state throughout its lifecycle.

Weekly analysis 8

This week there was a noticeable progress in my skillset as well as addressing frontend design problems and implementing Redux state management (Ighodaro N. 2023) in the project. One of the key milestones was on Monday when I took on a task assigned during our daily sprint meeting, focusing on enhancing the layout of gift cards on the frontend. As I worked to replicate the specified template accurately, I faced challenges with styles not aligning properly. Despite these obstacles, I implemented various subtle adjustments inspired by the figma design. These adjustments included refining padding, highlighting specific areas, and introducing hover effects. Furthermore, I streamlined the presentation of gift card items by eliminating unnecessary elements, thereby highlighting only essential information to users.

On Tuesday, I got better understanding into addressing frontend design issues. A major focus was on refining specific styling properties within the gift card array. One of the significant issue was the excessive spacing between each gift card, which necessitated a thorough examination of the component tree and subsequent adjustment. By making multiple changes within the gift card component and container tree, I was able to enhance the layout successfully, ensuring proper spacing and centering of the gift card array. The change from a display grid to a display flex approach led to achieving the desired layout outcome effectively.

As the week progressed to Wednesday, a new task was assigned to me following the sprint meeting - the implementation of a search filter using Redux. This task entailed the creation of various essential files, including reducers, action creators, and a combined reducer for the store. My initial steps involved establishing constants and defining functions within the reducer to manage the filter state. Additionally, I dedicated efforts to a separate user interaction component, enabling users to input values that were processed using the setFilter function from the reducer class.

Thursday's focus was on further enhancing Redux state management by introducing reducers for content or gift cards. Following a similar pattern as the filter reducer, I incorporated different action types and return logic for each scenario within the gift card reducer. Subsequently, I amalgamated both the filter reducer and the gift card reducer into a combined reducer. Making minor modifications to App.jsx and main.jsx, I integrated the Filter component and set up the Redux store before committing changes to GitHub.

On Friday, my attention shifted towards further improving the frontend by implementing Redux state management across additional files. Leveraging the Axios library (Axios, 2016) for data retrieval from the Mongo database (Tarud J. 2024), I focused on enabling a feature to generate and send gift cards to the backend through post requests. Establishing a dedicated Redux folder containing actions, reducers, and store files, I specified event triggers in the actions file, functions in the reducers file, and finalized the integration of Redux into the project with the creation of a Redux store.

The activities of this week presented invaluable learning opportunities, particularly in Redux state management and troubleshooting frontend design challenges. Looking ahead, my aim is to persist in refining the frontend design, exploring advanced Redux concepts, and making continued contributions to the project's overall development and success.

4 Discussion

Considering my initial situation, I have changed a lot professionally and personally. I have gained plenty of technical skills and soft skills. I've learnt how to react in professional scenarios and methods. I was able to find plenty of solutions ranging from the project management style such as scrum to specific technical things such as solving errors in my code, reading documentation, understanding the code better, debugging, etc.

This diary-based thesis has given me a great chance to reflect on my professional development as a software developer, it helped me recall the things I've learnt throughout my career, including tech stack I've learnt, and analysing later on what improvements could be made. Interesting things which I noticed during my thesis was the way I described my workday, describing routine, specific tasks related to work, problems I encountered, tickets and task which I've accomplished. The benefits were better understanding of my work by describing it in my own thoughts and words.

In my diary-based thesis, I utilized daily job analysis as a tool for understanding the complexity of my role as a software developer. By documenting my daily tasks, challenges, and accomplishments, I was able to analyze the specific requirements and responsibilities of my job more comprehensively. This process allowed me to identify patterns, areas of strength, and areas for improvement in my work performance.

Moreover, job analysis helped me gain insights into the broader context of my role within the organization and the software development industry as a whole. By examining the skills, knowledge, and competencies required for success in my position, I could plan my professional development efforts around it. This included identifying areas where I needed further training or skill enhancement to improve in my role.

To further develop my skills as a software developer, I'll be sure to have continuous learning which can include doing courses, cloud certifications and following best practices in software development. I can get more hands-on experience by working on my personal projects, contributing to open-source projects which will expand and deepen my technical knowledge. Another thing which will assist me in my career can be more networking as it will help connect to my peers which opens more ways of working, different opinions, ways of coding, etc. Last thing would be to improve my soft skills such as communication, collaboration, and problem-solving which can be enhanced

through teamwork, leadership roles, and interpersonal interactions. feedback on my code which will help me identify areas for improvement.

Sources

Ahtari A. 2023. Type–checking Made Easy: Exploring PropTypes in React. URL: https://blogs.perficient.com/2023/06/28/type-checking-made-easy-exploring-proptypes-in-react/ Accessed: April 29th, 2024.

Amazon 2024a. Amazon CloudFront: Content Delivery Network. URL: https://aws.amazon.com/cloudfront/ Accessed: March 5th, 2024.

Amazon 2024b. Amazon S3: Scalable Object Storage URL: https://aws.amazon.com/s3/ Accessed: February 26th, 2024.

Amazon 2024c. Getting started with AWS Lambda URL: https://docs.aws.amazon.com/lambda/latest/dg/getting-started.html Accessed: February 29th, 2024.

Axios, 2016. Getting started URL: https://axios-http.com/docs/intro Accessed April 18th, 2024.

Cypress.io, 2017. Comprehensive test automation guide URL: https://docs.cypress.io/guides/overview/why-cypress Accessed March 25th, 2024.

Ighodaro N. 2023. Understanding Redux: A tutorial with examples, URL: https://blog.logrocket.com/understanding-redux-tutorial-examples/ Accessed April 16th, 2024.

HashiCorp 2014. Getting started with Terraform. URL: https://developer.hashicorp.com/terraform/tutorials/aws-get-started Accessed: March 4th, 2024.

HuaChao Mao 2016. REST Client URL: https://marketplace.visualstudio.com/items?itemName=humao.rest-client Accessed 21st March 2024.

Mühlemann J i18next 2011. Official documentation, getting started with i18next URL: https://www.i18next.com/ Accessed: February 22nd, 2024.

Npm, 2010. npm Package: bcrypt. URL: https://www.npmjs.com/package/bcrypt Accessed: March 27, 2024.

Postman. Postman 2012, The API Platform https://www.postman.com/ Accessed: February 4th, 2024.

Prettier 2017. Why Prettier?URL: https://prettier.io/docs/en/why-prettier Accessed: 14th March 2024.

Sharma I 2022. Advantages of Refactoring in Software Development URL: https://www.tatvasoft.com/outsourcing/2022/09/advantages-of-refactoring.html Accessed: December 8th, 2023.

Tarud J. 2024, MongoDB: Advantages and Disadvantages your enterprise should consider URL: https://www.koombea.com/blog/mongodb-advantages-and-disadvantages/ Accessed: 29th April, 2024

The Open JS foundation, 2013. Documentation ES Lint pluggable Javascript linter URL: https://eslint.org/docs/latest/ Accessed March 25th, 2024.

Wieruch R, 2022. React Router 6: Private Routes (alias Protected Routes) URL: https://www.robinwieruch.de/react-router-private-routes/ Accessed: 19th February 2024.

Harband J, 2011. NVM SH GitHub repository: nvm-sh/nvm https://github.com/nvm-sh/nvm/releases Accessed: March 25th, 2024.

Appendices

Appendix 1. xxx