Project Exam 2

Holidaze – hotels in Bergen

Arshad Shakil

Word count

Summary: **99** | Main text:**2190**

Table of Contents

School of technology and digital media

1. Summary

This report shows the work done in the project exam "Holidaze". It shows some of the reasoning behind the work done. It is built using React and utilizes a UI framework. Designed the site at best effort and would have bought Tailwindcss if it were known that they have premade UI components bigger than bootstrap. Found most bugs and kept developing the site until the last day. Has potential to improve.

The site is deployed onto Netlify and utilizes an API hosted at Heroku.

The site can be found here: **https://vigorous-nightingale-785e07.netlify.app/**

Username: admin@admin.com
Password: Pass1234

Repo: https://github.com/Noroff-FEU-Assignments/project-exam-2-arshadshekel

School of technology and digital media

2. Body

2.1. Introduction

This report shows the work done in a project exam given by Noroff. Out of two options I chose to make a website that shows different hotels near Bergen. The site is to be called Holidaze and makes it easier for people to find a hotel.

In addition to making a Gantt chart, we must create style guides, prototypes, code the site in Reactjs/Nextjs, with an API backend which supplies the site with information. Towards the end of the project another student is to look over the code and give feedback.

As mentioned, I chose to make the Holidaze app because I would most likely not be able to find a client who I could work with. I chose the safest route.

2.2. Main section of report

This portion of the report contains the work done in the project. All aspects of the project will be mentioned in each subchapter.

2.2.1 Gannt chart

For the Gantt chart I chose to add all bullet points from the requirements of the assignment. This means doing research, building style guide and prototype, coding and review of the code. The Gantt chart is shown in appendix 5.1.

2.2.2 Likely users

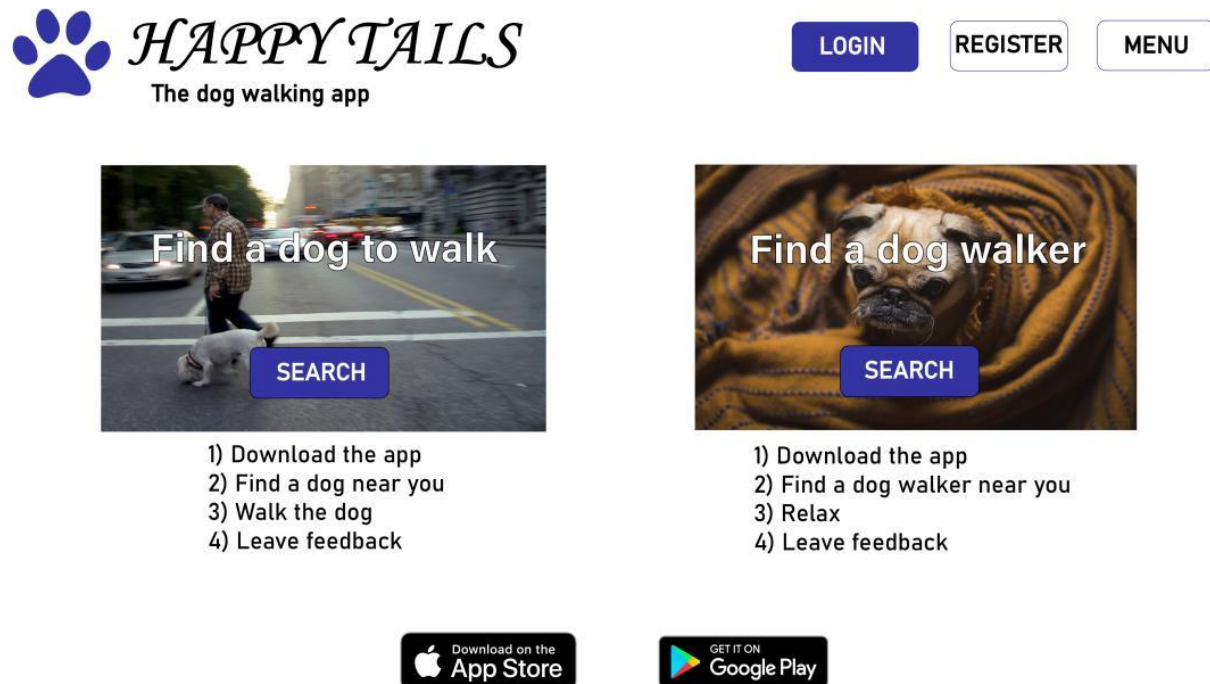Any person between 15 and 99 years who wants to visit Bergen.

2.2.3 Prototype and style guide in Xd

I made a simple prototype after my best effort. No, I did not search for inspiration, no I did not look for previous exams that had been delivered. I am stubborn in my way and I just try to build something that I like. The prototype is just as a small guide and to give me a sense of what I should do. The product might end up being different, which it did. I ended up scrapping the very round corners and toned it down. I also skipped making a prototype for different viewports because this is something I just figure out as I code.

I also made a simple style guide in the same Xd file. The font sizes were added after I figured out how I want them on the home page. The rest of the font sizes are all standard and I use h1 to h4.

I did not put much effort into the design, because I know that I have limited skill as a designer. I am a developer at heart, and even though I appreciate the focus on design from the course, I am not going to put lots of effort into it. Its not because I think design is bad, but because I will either use templates if I work as a freelancer, or if I am in a company someone else will handle it. Let me ask you a question: Remember the "Happy Tails" mock-up we were supposed to deliver? A site where people can find and walk dogs? Well, this is what I ended up making:

Does it look like I have any idea what looks pretty? I have no aesthetic sense in real life either. I feel like I have progressed a lot since that time and I am happy with my effort despite it being a bit low. My focus for this exam was learning how React works.

2.2.4 Coding

For this project I chose to use React as the framework for JavaScript. For the UI I used react-bootstrap [1]. I regret learning that there were people in the class who used Tailwind components that their employers had paid for. But still I did my best with bootstrap. I also used SCSS. My goal was as I said to learn how React works and how I can do as much as possible with it.

The project is coded using functional components. I felt that was easier to grasp. I also got more comfortable with useState and useEffect. However, I should have done more of my homework. The whole project was a more hands-on approach where I sit and try to break code. That is how I learned programming during my bachelor studies. The very last day I came up with the idea to check Scrimba and the advanced React courses. I was glad that I had managed to utilize most of the components. The only things I regret is not learning about useReducer earlier. And the use of previous state.

I imported icons from react-icons [2], and I used react-bootstrap-typeahead [3] to code the search bar. Working with the typeahead library made my head hurt. At first, I could not make it redirect when the options appeared, and I clicked on them. So, I wrote hacks to be able to do so. I added a button as submit. Then I just realized I had used onClick the wrong way, so it made my code better. I also adjusted the search bar the last day, which can be seen in the exam repo. Also, the onkeyup only triggers when the filtering is done.

I utilized modals and toasts to make things a little more challenging. I did not bother making my forms better, I added a max width. As I said I want to be able to make things work. I downgraded the yup and react-hook-forms so they used a different syntax for form validation.

When creating or updating a hotel yup appended the file also in the data object. So, I quickly learned I needed to delete that entry, and then append the file in addition. I also added check for picture and validation.

The admin dashboard was severely limited, but it got the point through. I wish I had tailwindcss so I could make a beautiful dashboard. They have a component for that too. Anyway, I feel my dashboard is good enough.

As for message and enquiries I kept it simple. The point is not to have loads and loads of fields. Like I said I just wanted to make sure I knew how to use the framework and libraries. I also set a limit on myself or else I would be adding features forever and forever. Simple is good enough for me. I added so that public can only post but not get the messages without validation.

In the hotels page I added filtering, so I knew how it works too. I added the search bar after I had finished the filtering functions. And it was the day before delivery. I would have spent time trying to add filtering of hotel cards based on the search input too if I could.

As for the cards I made sure that the hover effect only colours the cards on higher resolutions. It turns out that om mobile it suddenly can colour the card. So, I added a media query for that

The login and logout work as intended. I use the useContext hook to get the auth state. Like I mentioned earlier I spent time on Scrimba the very last day and learned about useReducer and Redux. So, knowing how useContext API works, I know can do small to medium sized projects.

The night before delivery I figured I be smart and look at how to implement slugs instead of id. Ended up making a mistake in the strapi config. Everything looked good until I tried deleting a hotel. Turned out it deleted the entire array. And I was unable to log into the admin section to add new. So, I had to restart and rebuild my API from scratch. Luckily, I had saved the pictures, so it was easy to fill in the rest. I also made a google API key and embedded a map. It was surprisingly hard to figure out which service to connect the API key to, but it worked in the end. Maybe I was too impatient who knows. I am anyway awfully glad I made it work.

I deployed to Netlify and used the exam repo. I also made sure to add CI= so that it continuously follows Github for changes.

2.2.5 Hotels for the project

At first, I thought I'd add many hotels from Bergen and just use the official pictures they put out. But then I remembered that its illegal so I looked for creative common's pictures of hotels in Bergen and added each I could find. I ended up with 8 hotels. I hope that is enough. I also found the same hotels on google and used the info there to fill out address and the description.

2.2.5 Code Review

After talking to MJ about it I was told I could skip the code review. This is due to me not wanting people to copy my code or ideas. But looking back at my code now I should have made a component for the showing of Modals. And I should have had a form for contacting Holidaze, which gets implemented in the contact us page. Just to stay consistent. Also, I have failed miserably at BEM (just remembered it was a requirement). I did however give the classes meaningful names. I also see now that I have used named imports instead of default imports when importing from libraries.

I realize I could have structured the files better. I also realize I could have crated even more components to make my code better. But the size of the project is not that big. So, I did not see the point. I did also not set Proptypes because I had it in my mind what type of each variable is. Maybe I should have added it, but I am good at reading the console and error logs. So, I quickly fix my problems.

2.2.6 Bugs

I have been very methodical in testing my code and looking at the console while trying different things. At one point I used to have problems logging in after logging out unless I refreshed the site. Now it seems to be working OK. I have tried my best to keep bugs out. There as a bug while testing on localhost; buttons would not respond to clicks and not show cursor when hovering over the button. This annoyed me the most because it did not show in console. Hopefully, it is gone now.

Introducing slugs instead of id for show in address bar introduced a lot of problems because I did not know how to fetch it correctly. Now that I got that in order it works well enough. I also noticed a bug when clicking on the menu link, that gives it an active state. But if you get pushed to another page, they both show as active. That was fixed by setting the state of active to false on each link by default. I also noticed that clicking on the link while on mobile did not close the menu. So, I added some functionality for that too.

All in all, I think my code is good enough. And I hope I do not get more bugs.

UPDATE: I was making my preparation to deliver and read the brief one more time. Turns out I missed the part with the list of enquires being refreshed. So, I had to

code it. Also, I found out why the buttons broke. I applied a fixed-bottom class which blocked the inputs of the mouse. I made the class appear only when firing toasts.

UPDATE 2: I managed to show enquiries by the latest one received. I also made a setInterval that refreshes the list of enquiries. I added styling to show which messages have been opened. I also added som icons to make it look prettier. As for the content of the enquiry I slacked because I don't feel I need to practice how to style things or present them on the screen.

UPDATE 3: ITS 2353 AND RENAMING THE TITLE OF HOTEL DURING EDIT BREAKS LINKS

2.3. Conclusion

I am satisfied with my effort in this exam. My main goal was to decode how React works and make it work. My prototype was simple, but it gave me an idea on how I want to make my site. I know my major skills are not design, however I will try to improve for the portfolio course. All in all, I feel I have delivered a decent project. I am glad I got to get the chance to study at Noroff, despite me loathing design at first. And despite me not giving 110% on the design or documentation front I still feel I have managed to make something that looks OK.

3. References

[1] "React Bootstrap," [Online]. Available: https://react-bootstrap.github.io/.

[2] "React Icons," [Online]. Available: https://react-icons.github.io/react-icons/.

[3] "React-bootstrap-typeahead," [Online]. Available: https://www.npmjs.com/package/react-bootstrap-typeahead.
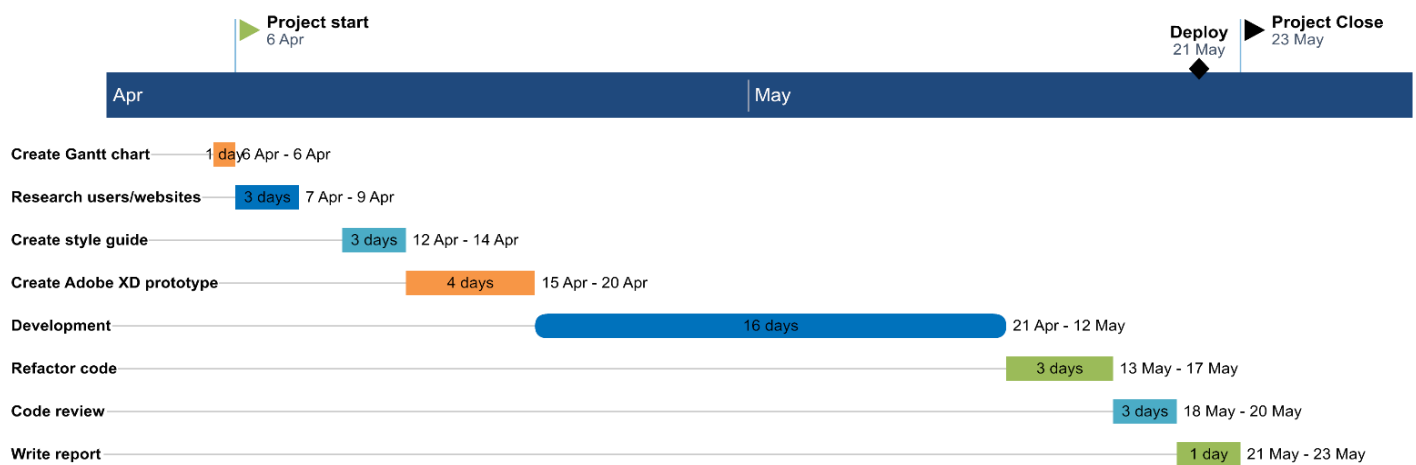
4. Acknowledgements

development. Looking at the job ads it seems we have covered a lot of subjects. And its good that you keep updating the curriculum.

And finally, I would like to thank Google for helping me find documentation.

5. Appendices

5.1 Gantt Chart

Gantt chart for project