



Noroff

School of technology
and digital media

Technical Report

Holidaze

Marek Fischer

Word count

Summary: 108 | Main text: 1443



Table of Contents

1. Summary.....	3
2. Body.....	4
2.1. Introduction.....	4
2.2. Main section of report.....	4
2.3. Conclusion.....	4



1. Summary

This project exam required the creation of a website called Holidayze for tourists in Bergen to find places to sleep. The project was planned using gantt charts, a basic style guide and an adobe XD prototype and coded using Typescript and React. The back-end was done using a vanilla Strapi install and a few external libraries were used for the rich text editor and google maps embedding. Some refactoring was done using the feedback of two peers and a few jest tests were included. The website features search and suggestions, inquiry modals and an admin page with the ability to change most of the content on the website.



2. Body

2.1. Introduction

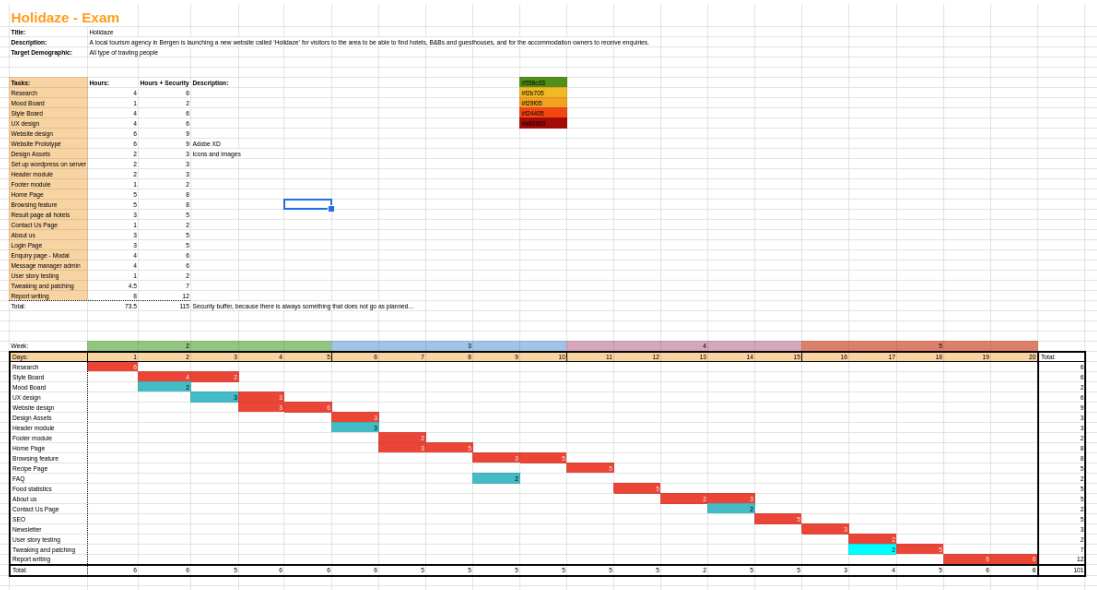
The goal of this project was to create a website called Holidaze with the goal of giving the ability to find hotels in Bergen and send messages to the establishments and for the owners to manage their establishment and respond to inquiries through an admin panel. This reports goes through the process I followed while creating this website.

2.2. Main section of report

The first step of every project is planning, and these are the steps I did to plan this project in order to manage time, and make it easier for me to have progress consistently as making a list for the future you and then following a list of task is easier than to having to have an overview of the whole project at any time in order to know what to do next. I started this process by first choosing what option I was going to choose. I am currently working as a full time full-stack developer at Acos AS since the first of October and I tried to see if I could in some way combine the assignment with the work I already do. We are currently on the verge of releasing a new version of our CMS product that will go live in a couple days at the time of writing this and I was allowed to spend some time on Typescript React components that is useful for both my work and my assignment. However, this was not enough to satisfy the requirements for the “real world client” and my time is very limited, so I decided to go with option 2 and make a hotel booking website.

I started by making a list of features I wanted to include in the project, and made a Gantt chart.

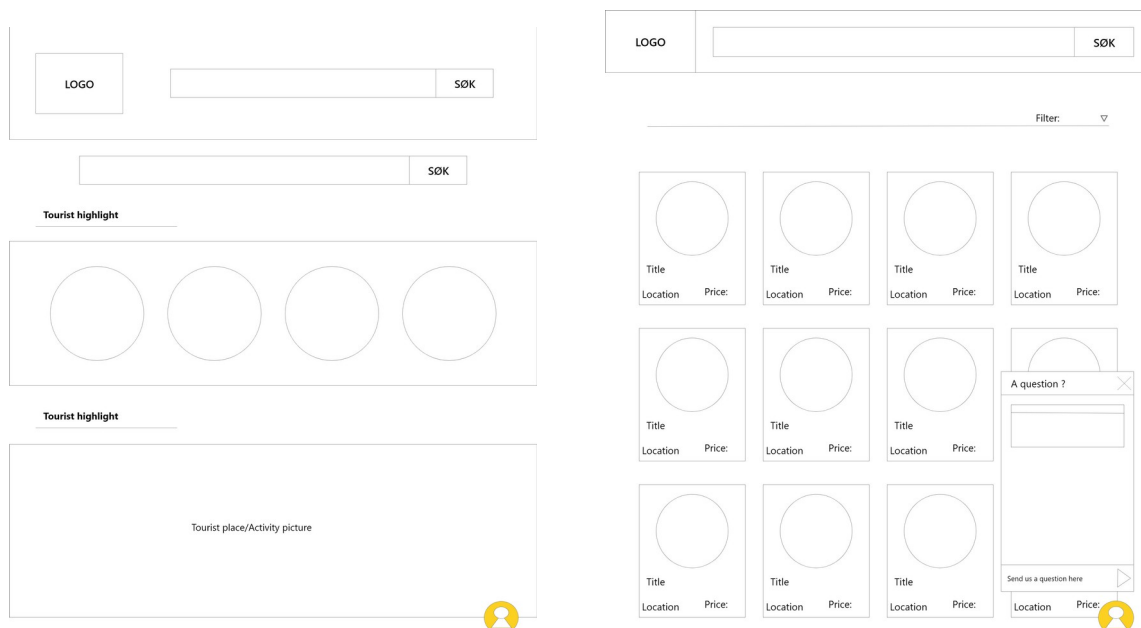




The numbers was meant to be the amount of hours for each task, but calling them story points would be a lot more accurate.

I also ended up using trello a lot to keep track of my tasks which helped greatly.

I then proceeded to make a prototype on Adobe XD. I first had to install windows on a second partition in order to be able to run the program.



I had a few ideas that I wanted to implement. The first was a the double search bar, but the original idea was to have one search bar integrate in the nav bar on scroll. The second was the inquiry pop up button in the bottom right corner.

The next step was to write the code. I first decided to setup a headless Wordpress installation, as I already have a Strapi deployed on my server. However, when I started working with it I realized I had no idea how to add custom fields into the rest api, and using the existing post and page concept seems completely ridiculous. There probably exists an appropriate way to use Wordpress in this way, but the software is so bloated that I decided it wasn't worth wasting the little time I had on it.

I then moved over to Strapi and decided to work some proxy magic on my server to make it work. The API is currently online at <https://holidaye.deepvertic.com>.

I was able to code three components at work while working on this project. The search bar, the modal and the button component. These components were further modified a lot since then to accommodate for new features that will go live soon on most public government websites, but the more rudimentary versions were enough for this project.

I started by implementing the search component into the navigation component. I ended up duplicating the search bar for the home page because of a lack of time, but I felt like it still made sense to have it in the middle of the home page too as searching is on of the first thing you would do on such a website. The actual search API calls were easy to implement and simply tests if the search string is included. I added support for pagination and suggestions that can be navigated through using the keyboard tab and arrow keys.

I then started working on the contact system. I tried to make the contact system and inquiry system separate but similar enough so that one component could do both. I ended up with a contact button on the home page, and an extra "ask the hotel" button when going to the page of a hotel. This way the client may contact the website, but easily also send a message to the establishment, and all that lies in the same content table with an extra field for the hotel id. I ended up using the modal



component to render the contact pop up. I found a way to easily render components at the bottom of the body tag by programatically inserting it in the right place, wherever the actual component is inserted. This makes it much easier for screen readers to pick up the fact that there is a pop up instead of having to fight with z-indexes etc.

I then moved on to the login page and then to the admin page. I went for a one page accordion/tabs design as there isn't much content to manage. I used many bootstrap components to make the admin page, and used a rich text editor called tinyMCE for the hotel page content. I reused the modal component for several alert pop ups for feedback and deletion. I then added two tabs, one for the contact forms and another for the hotel inquiries. I added the ability to mark the messages that has been processed and the ability to choose from a drop down the hotel you would like to manage.

The next page to code was the actual hotel page where I included the google maps API to generate maps based on the location field. I also added the content as dangerouslySetHTML which places the burden on the hotel page editor to make that part of the website look good. This is made easy with the ability to paste pictures, resize and modify the text size and color, and a lot more.

The next step was a bit of refactoring. I cleaned some of the code a bit, formatted things more neatly and added a few more checks for potential undefined data. This is also where I asked two friends to review my code. One of them uses primarily flutter for his web applications and the other was a PHP developer, but they still managed to give me some constructive critique. A lot of it was inconsistent naming cause by the fact that my work place has a specific coding standard based on older technologies and I have my own habits when coding on my own projects which I find more readable in my opinion. I also redid some of the logic of some components where I had used forwardRef in places where simple callback functions did the job well. They also helped me spot a few bugs as for example the page scrolling down when pressing the arrow key to cycle through the suggestions when searching.



I later also realized that my error checking is subpar and adding some loading skeletons would make the website look a lot more polished.

This concludes my process of making this website.

My time was very limited and I ultimately only got to put in 25 hours of active coding, excluding the three components I did at work, because of the fact that I am a dad who has a full time job as a full-stack developer, so it has been a rough couple months finding enough time to do this assignment so there is a lot I would have liked to improve.

This includes:

1. Better error handling with visual feedback to the user.
2. Emailing system using an Email Strapi plugin.
3. Further refining the focus control and keyboard navigation.
4. Adding some better examples of hotel content.
5. Using redux, and relying less on bootstrap.

2.3. Conclusion

After a couple of months the website is now working well with all the core functionalities implemented. Some components were made in the context of my full-time job which let me kill two birds with one stone on some occasions. Some basic planning was done in the beginning, some design and prototyping too. The hotel booking website was coded in around 4 weeks and was tested by friends and the code was reviewed and refactored slightly before deploying the back-end on my personal server using proxies to keep it in developer mode while still being online. This concludes my project.

