

# Holidaze – Exam

## Innholdsliste

|                              |    |
|------------------------------|----|
| Introduction.....            | 2  |
| Planning.....                | 3  |
| Goal.....                    | 3  |
| Flowchart.....               | 4  |
| Gantt.....                   | 4  |
| Tools.....                   | 5  |
| Angular.....                 | 5  |
| Express.....                 | 5  |
| Mongodb.....                 | 5  |
| Mongoose.....                | 5  |
| Jsonwebtoken.....            | 5  |
| Bcrypt.....                  | 5  |
| Design.....                  | 6  |
| Wireframes.....              | 6  |
| Styletile.....               | 7  |
| Prototype.....               | 8  |
| Code.....                    | 9  |
| Frontend.....                | 9  |
| Backend.....                 | 11 |
| Feedback - Jonas.....        | 12 |
| Overview and Conclusion..... | 12 |
| Github.....                  | 13 |
| References.....              | 13 |

# Introduction

The website holiday is for visitors to be able to find hotels, B&B and guesthouses. And for the accommodation owners to receive enquiries.

The website needs the following:

- **Visitor**
- Homepage
- Search bar that filters hotel names
- Result page with all hotels
- Hotel specific page
- Enquiry page
- Contact Page
- **Admin**
- Login
- Incoming Enquiries
- Messages from contact form
- Create new establishments

# Planning

## **Goal**

With the time available I wanted to try expand on the project and I did it like this:

- All accommodation owners get the title of “Hotel” for simplicity.
- A hotel account can create multiple hotels because the person might own multiple properties.
- Every hotel can contain multiple rooms that can be rented out at the same time.

- **Visitor**

- Homepage
- Search bar that filters after hotel names
- Alternative “advanced search” for rooms
- Hotel specific page
- Room specific page
- Register
- Login
- Contact Page

- **User**

- Enquiry for hotel rooms visible when viewing a room.
- “My Page”
  - View orders

- **Hotel**

- “My Page”
  - Create Hotel
  - Create Room
  - View Reservations

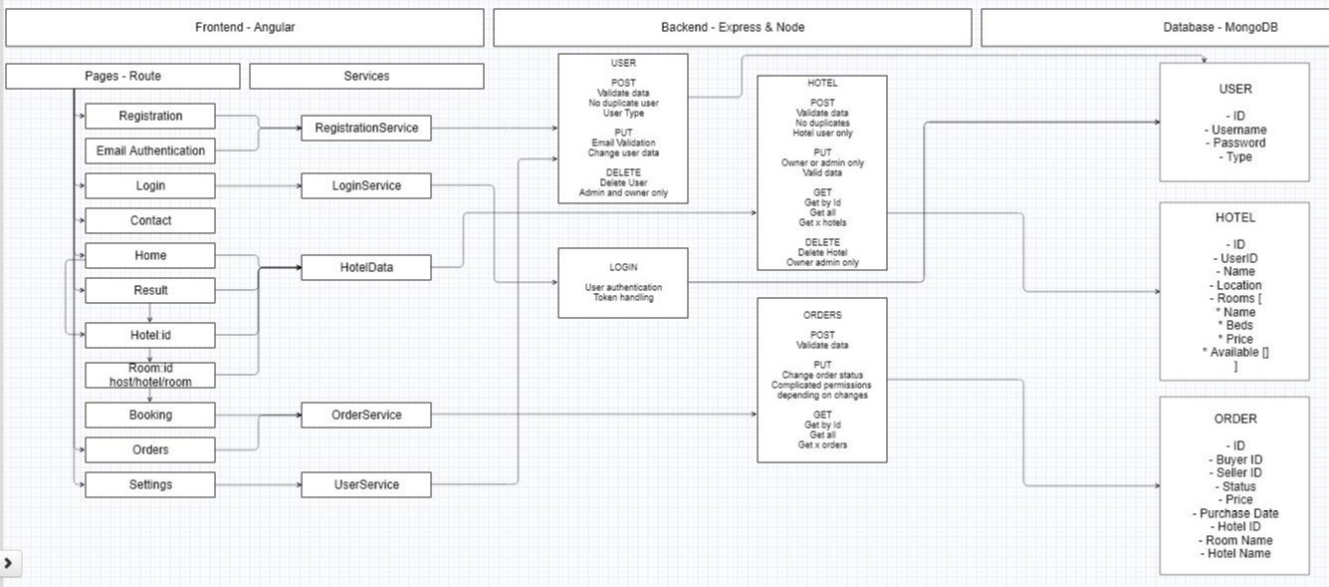
- **Admin**

- “My Page”
  - View contact submissions
  - Can also do whatever user and hotel accounts can do.

Wanting to create user registration and login require it to be stored somewhere, this is not supported by the backend provided during the project. Therefore I would have to create my own solution.

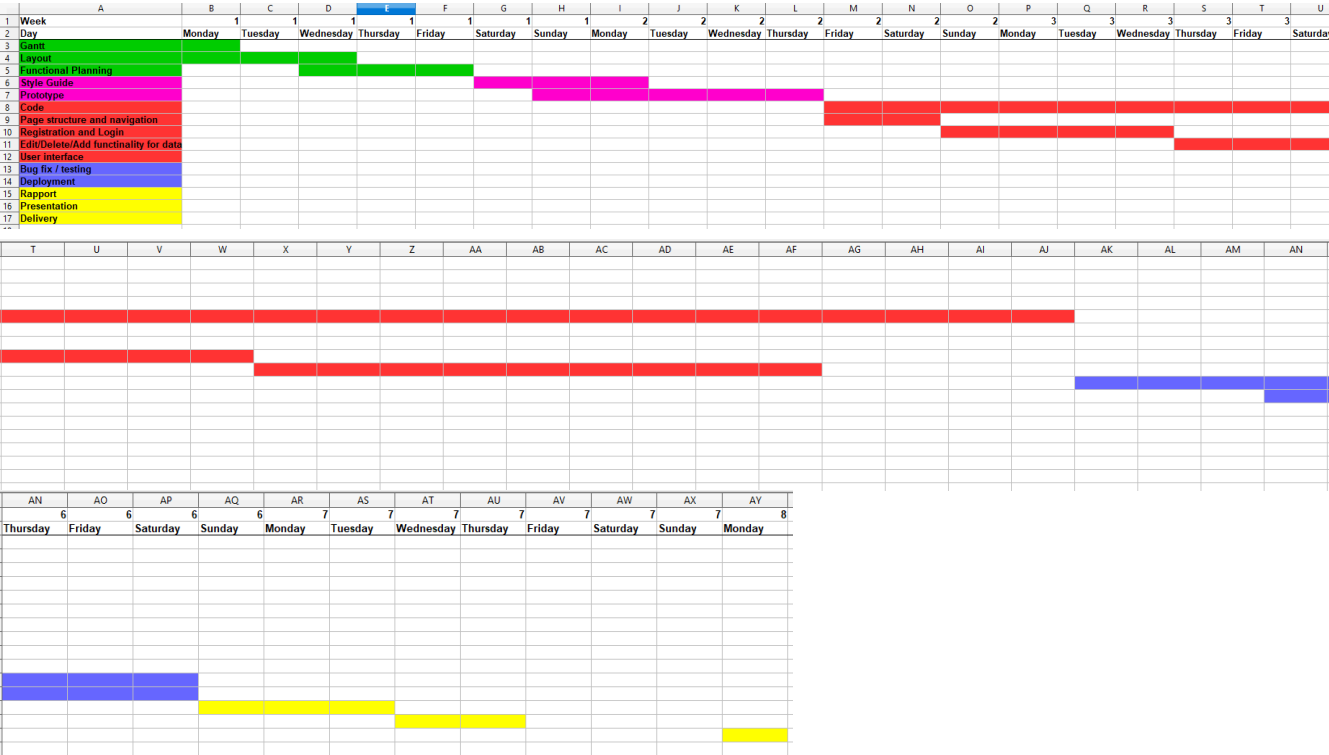
Flowchart

Below is a overview of how I expected the frontend – backend and database to communicate with eachtoher.



Gantt

Below you can see how I expected to be using my time during the project.



# Tools

## ***Angular***

For the frontend part of the project angular was used. Under this framework I used typescript and scss. Did also use the form tools and validation built into angular. There were also multiple services dealing with different types of data. Token is stored in the environment json file when users is logged in.

## ***Express***

Using express to run the backend server. The backend is split into public and src. Public folder includes the frontend build that get sendt to users connecting to root.

Src folder contain middleware, models, mongodb and routes. This folder deals with all the endpoints for the backend.

## ***Mongodb***

Mongodb is the database that is used for this project and store 4 types of data: Contacts, hotels, rooms and users. Contacts contain data about submissions from the contact page, hotels contain data about hotels, rooms contain data about rooms within any given hotel, and users contain data about any registered user on the site.

## ***Mongoose***

Mongoose is the tool I use for communicating with the mongodb.

## ***Jsonwebtoken***

Jsonwebtoken is used in order to verify if a user is logged in or not.

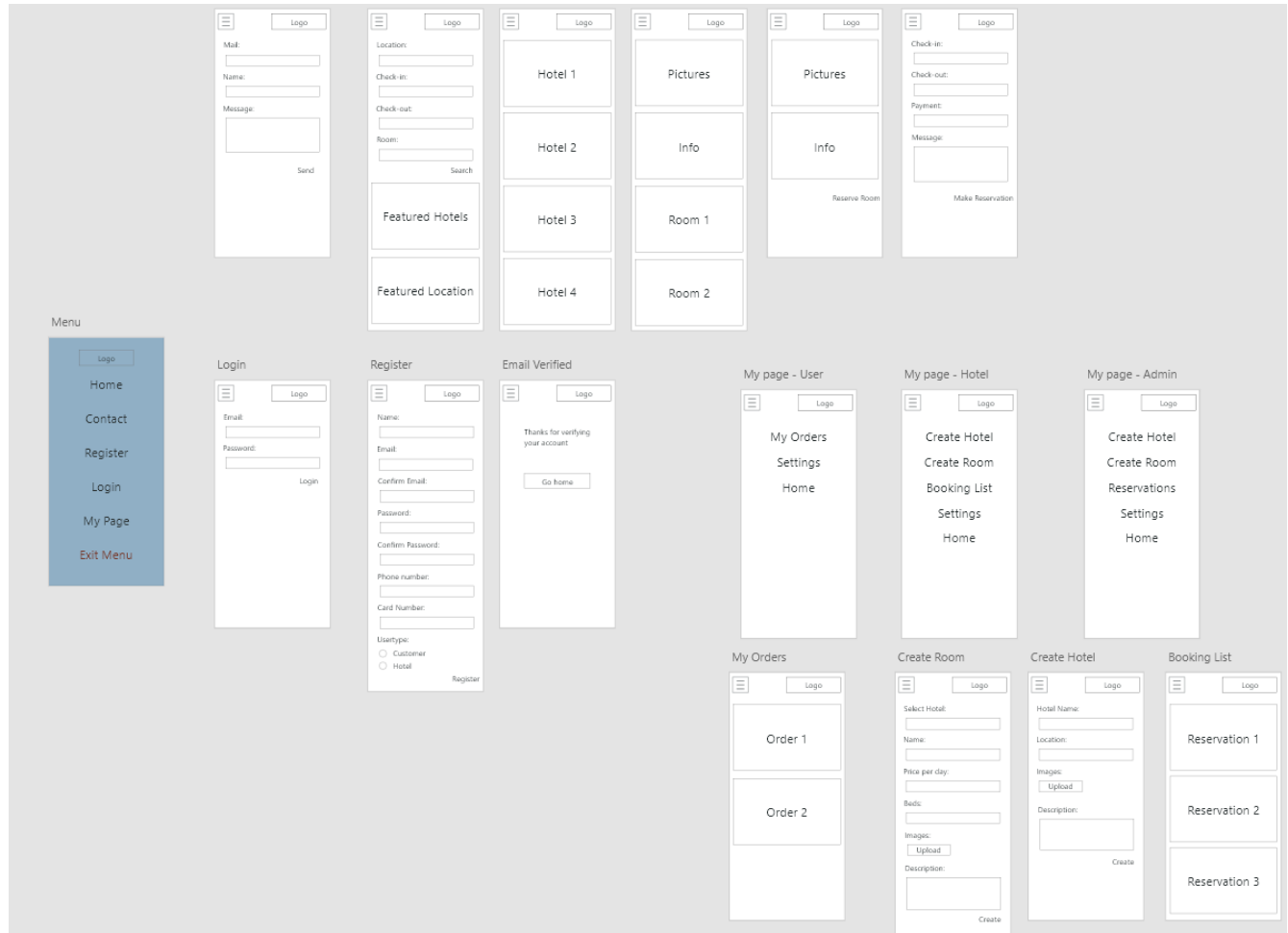
## ***Bcrypt***

Bcrypt is used to encrypt password before adding it to the database and comparing passwords during login. To make user information more secure.

# Design

## Wireframes

When creating the design for the website, there was first created wireframes in mobile format to get an idea of how the website could be structured.



## Styletile

After having a structure for the website I created a styletile for how things would look like on the website.

Font: Roboto

Paragraphs will be written with thin roboto font.  
With font size of 15px.

**Header 1 - H1 size - Roboto Medium**

**Header 2 - H3 size - Roboto Regular**

**Header 3 - H5 size - Roboto Regular**

Warning - Error Message - 15 px - Roboto Thin

Standard

Warning

Call to Action

*Holidaze*

Input

Favicon

*H*



#8DD4FF



#336B87



#2A3132



#763626



#FF9F9F

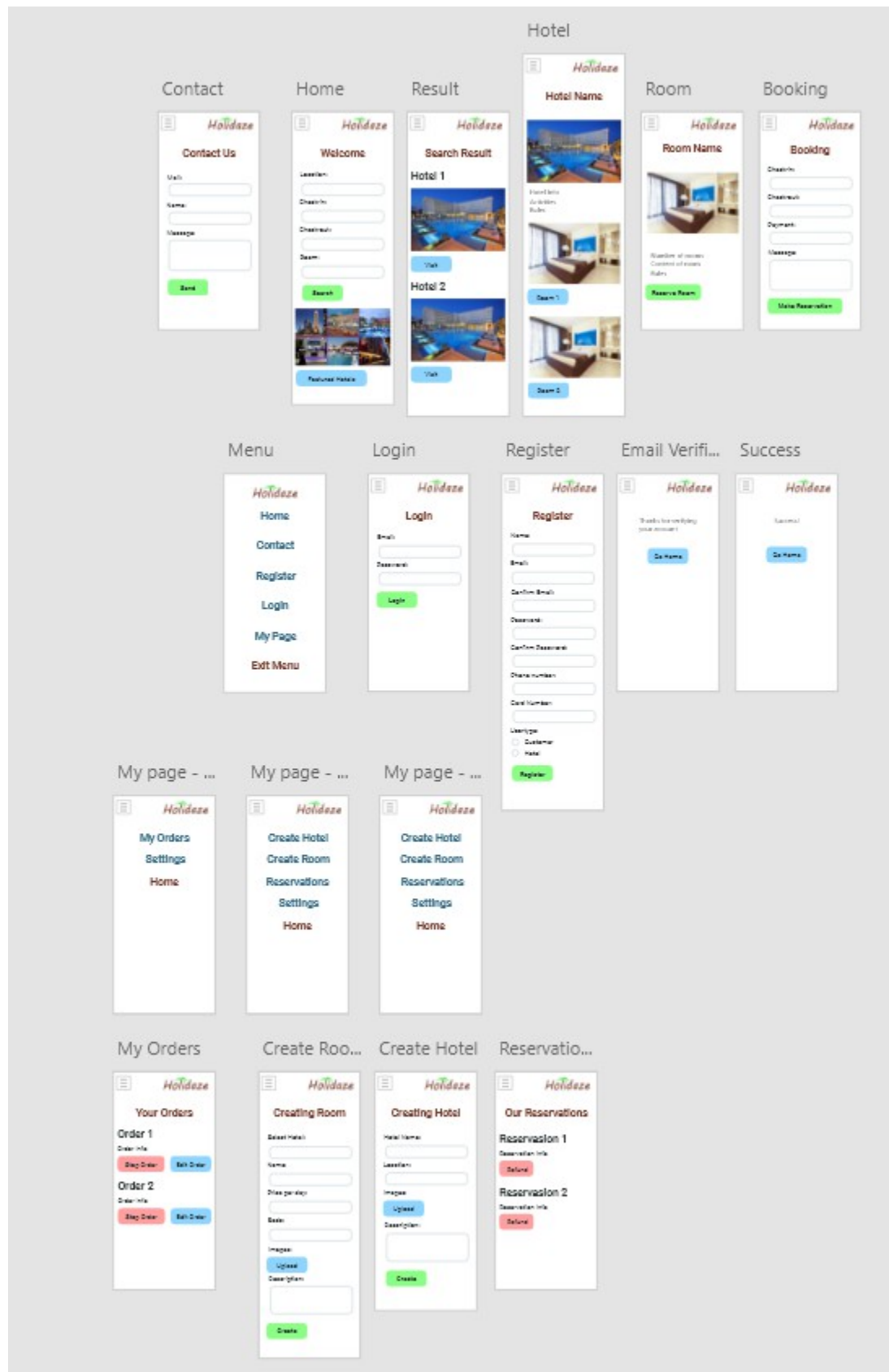


#FC3232



#8BFF85

# Prototype





# Code

## ***Frontend***

### **Navigation**

The navigation will display home, contact, login and register while not being logged in.

This will change to home, contact and <username> when logged in. When clicking on your own name in the navigation you will be directed to “my page”

### **Contact**

Contact page has 4 required fields in order to send a message, you can read these messages if you are logged in as a admin. (admin users can not be created through the user interface, this is a reserved user type)

### **CreateHotel**

If you are logged in as a hotel and you are in “my page” you can select the option to create a hotel. Every hotel user can create multiple hotels that are located at different locations.

There are 3 required fields, image url is optional.

### **CreateRoom**

If you are logged in as a hotel and you are in “my page” you can select the option to create a room.

You can create multiple rooms for every hotel you have created. You can only create rooms for your own hotels. From the dropdown you can select the hotel name for the hotel you want to assign the room to.

### **Home**

On the home page you can search for hotels or hotel rooms.

If you search for hotels you will automatically filter through hotel names as you type, if you change the search to hotel rooms you can filter hotel rooms after multiple criteria, but you have to click search when you are done.

### **Hotel**

When selecting a hotel you get directed to the hotel page. This page displays some info about the hotel and what rooms that can be rented within the hotel.

## **Room**

When selecting a room you get some information about the room and if you are logged in as a user you have the option to reserve the room. Option to reserve is not visible when logged in as hotel.

## **Login**

Here you need to login with the username and password you created. If the user information is correct you should receive a json web token and show as logged in. If you refresh the website you are required to login over again.

## **MyPage**

This page will give access to different pages depending what kind of user you are logged in as.

## **Order**

Here you get an overview of all the rooms you have ordered as a user. The information displayed is very limited, and this solution could be better.

## **Register**

On this page you can register a new user or hotel depending on what you want to accomplish on the website.

## **Reservations**

Here you can get an overview of all the reservations done on your hotels. This is just an option for hotels. The information displayed is very limited, and this solution could be better.

## **Contact service**

Handles all the data related to the contact form.

## **Hotel service**

Handles all data related to hotels or hotel rooms.

## **Login service**

Handle the process when a user try to log in.

## **Register service**

Handle the process when a user try to register an account.

## ***Backend***

### **Root**

When connecting to root you will be directed to the home page of the website.

### **Middleware**

There are some protected endpoints in this project. This middleware check that user provide a valid token. If there are no valid token the request is rejected.

### **Contact endpoints**

These endpoints handle data related to contact form on the backend.

### **Hotel endpoints**

These endpoints handle data related to hotels on the backend.

### **Room endpoints**

These endpoints handle data related to rooms on the backend.

### **User endpoints**

These endpoints handle data related to registration, login and users on the endpoint.

# Feedback - Jonas

I did get some feedback from Jonas on this project. He found a bug when trying to open a hotels page and this has been fixed.

He also gave some feedback on the home page not being as intuitive as it should be, this could still be done a lot better.

# Overview and Conclusion

This project has required planning, design, coding and bug fixes. There is still many things that can be done in order to improve the project.

When planning the project I created a gantt chart for how to distribute time, set goals for myself and created a flowchart to have a better understanding on how the website will function before I started.

After having a plan I started designing the project. I created wireframes first, then I created a styletile and used both as inspiration for a simple mobile prototype.

After having some idea of what I was making and how it was going to be designed I started to create the backend and frontend in parallel. Starting with the login and registration features and less complicated features as time went on. A lot of time in the beginning was used to try make the backend as robust as I could for my first ever attempt.

Getting close to the end of the project there were simply just some features I did not have time to implement and even some features that I am not happy with and wish I have done differently, but I am happy with what I have been able to provide anyway.

# Github

You can clone the project from the following github repo:

<https://github.com/Hortasha/Holidaze>

## References

Angular.io, *documentation (form validation)*, Available at: <https://angular.io/guide/form-validation>

Mongoosejs.com, *documentation*, Available at: <https://mongoosejs.com/docs/>

Youtube.com, *Academind Channel*, Available at: <https://www.youtube.com/watch?v=h4A0-53Olm4>

Fonts.google.com, *Roboto*, Available at: <https://fonts.google.com/specimen/Roboto>