

Project Exam 2

Project



🕒 Brief (../project-exam-2/brief.html)

🕒 Marking Criteria (../project-exam-2/marking-criteria.html)

Project Exam 2 brief

Table of Contents

- Goal
- Brief
- Requirements
- API
- Approved Resources
- Delivery

Goal

To take the skills learned over the last two years and take on an extensive project where the finished product should reflect the candidate's general development capabilities, in addition to visual and technical skills.

Brief

A newly launched accommodation booking site called Holidayze has approached you to develop a brand new front end for their application. While they have a list of required features, the design and user experience has not been specified. Working with the official API documentation, plan, design and build a modern front end accommodation booking application.

There are two aspects to this brief: the customer-facing side of the website where users can book holidays at a venue, and an admin-facing side of the website where users can register and manage venues and bookings at those venues.

Requirements

All API functionality is managed by an existing application. This project only covers the front-end application for the API.

API

The API you are using for this project can be found under Holidayze in the Noroff API documentation [🔗](#).

Resources

- API Documentation [↗](#)
- API Swagger [↗](#)

User Stories

The client has specified the following requirements in the form of *User Stories*:

1. A user may view a list of Venues
2. A user may search for a specific Venue
3. A user may view a specific Venue page by id
4. A user may view a calendar with available dates for a Venue
5. A user with a `stud.noroff.no` email may register as a customer
6. A registered customer may create a booking at a Venue
7. A registered customer may view their upcoming bookings
8. A user with a `stud.noroff.no` email may register as a Venue manager
9. A registered Venue manager may create a Venue
10. A registered Venue manager may update a Venue they manage
11. A registered Venue manager may delete a Venue they manage
12. A registered Venue manager may view bookings for a Venue they manage
13. A registered user may login
14. A registered user may update their avatar
15. A registered user may logout

Technical Restrictions

The company CTO has set the following technical restrictions:

1. Must use an approved JavaScript Framework
2. Must use an approved CSS Framework
3. Must be hosted on an approved Static Host
4. Must use an approved Design Application
5. Must use an approved Planning Application

Required Links

The Product Owner has requested links to the following:

1. A Gantt chart for project timing
2. A design prototype
3. A style guide
4. A kanban project board
5. A repository link
6. A hosted application demo link

Approved Resources

This list covers libraries and services that have been vetted by the company and approved for use.

JavaScript Frameworks

- React (>16)

CSS Frameworks

- Bootstrap (>5)
- Tailwind (>3)
- MUI (>5)
- Styled Components
- CSS Modules

Hosting Services

- GitHub Pages
- Netlify

Design Applications

- Adobe XD
- Figma
- Sketch

Planning Applications

- Trello
- GitHub Projects

Delivery

Include the required links in the Moodle delivery window using this template format (delivery-template.html).

All final changes must be merged into the default branch `main` or `master`. Other branches will not be checked.

Ensure that the `readme.md` file describes your project thoroughly, including how to setup and run the project locally and any special instructions for testers.