

# FlowrSpot

## Assignment Info

You are going to make a working version of a product called FlowrSpot. It is used for flower spotting while hiking, traveling, etc. Users can check out different flowers, their details and sightings, as well as add their sightings. You will not need to implement the whole product. Carefully read the instructions. The assignment is doable in a single workday time, but you can spend more time on it if you need.

## Project Designs and Assets

**Designs:** The project design is hosted on Figma.

<https://www.figma.com/file/94PDPTLqq2qBnMlmb7WTSU/FlowrSpot-Web?node-id=0%3A37>

Custom font: The font used for the project is Ubuntu and Montserrat.

**API: Documentation can be found on:**

<https://flowrspot-api.herokuapp.com/doc/index.html>

Note: this assignment is meant to mimic a real life assignment which means the documentation is not perfect and the designs are not completed 100%. Do the best with what you have.

## Instructions

This app will represent your level of expertise in building scalable web apps and we trust that you will solve it by yourself. You should be able to explain the choices that you made while planning and developing this app.

## General requirements

Use the React library (this is a React frontend assignment after all)

Use a state management system

Implement basic routing

Add a short README that explains how to build and run the project (and any other specifics if present)

Write at least one unit test and one snapshot test testing one of the required components

The project must have a local git repository (do not use public repositories to push the code, you will submit the assignment as a clean compressed archive that has a .git folder in it). Write clean commit messages, so it is clear what changes they contain.

Try to use common sense when implementing features - some are not specifically noted, but are necessary for the application to work as intended (e.g. close button on modals).

Required app elements style has to match the provided designs. Modals don't have to be responsive.

Do a last sweep over the code and clean it up before sending it in.

Required app elements

- Navigation:

Has to size nicely on any width

Mobile design is a bit different - check provided designs

- Home screen:

Implement only list of flowers:

List has to size nicely on any width

Get data from appropriate API endpoint

Pagination is not needed, list all returned in basic call

- Signup flow + API binding:

Signup modal with form and validation. Form inputs are:

Name

Surname

Date of birth

Email

Password

Signup success modal:

On successful signup display a message saying "Congratulations! You have successfully signed up for FlowrSpot!" with an OK button

Clicking the OK button opens the Login modal

- Login flow + API binding:

Login modal with form and validation. Form inputs are:

Email

Password

Login success modal:

On successful login display a message saying "Congratulations! You have successfully logged into FlowrSpot!" with an OK and a PROFILE button

Clicking the OK button closes the modal

Clicking the PROFILE button opens the Profile modal

- Profile modal window + API binding:

Use the placeholder image in the designs as a hardcoded user profile image

Show user data returned from /me API endpoint

- Authentication based elements visibility:

Logged out:

Show Login / New Account buttons in the navigation

Hide "favorite flower" stars in the top right corner of the flowers grid

Logged in:

Show user name and profile picture in navigation bar instead of Login / New account buttons

Show "favorite flower" stars in the top right corner of the flowers grid

Good luck!