

# Technical test @eCapitalio

## Project: BikeMyWay

Cities are complicated. BikeMyWay helps the user navigate in Paris using an eco-friendly, large-scale, public bicycle sharing system: **Velib'**.

Bikes can be picked up at any station near the user's departure point and dropped at another near the user's arrival point. Stations are located across Paris and in some surrounding municipalities.

BikeMyWay is a very simple web app, the user **does not need** to create an account to start using it. The user can simply input his departure and arrival points and the app will indicate the closest stations where he can pickup and drop a bike.

## Tech stack

### APIs

It is recommended to rely on these APIs, but **you can use an equivalent** if you feel more comfortable with it.

- Velib' API for stations data: <https://www.velib-metropole.fr/map>
- Google places autocomplete API: <https://goo.gl/f5BWbZ>
- Google maps JS API: <https://goo.gl/HJK8no>
- Google maps directions API: <https://goo.gl/KT8YZd>

### Mandatory

- React

### Suggested

We suggest those libs but **you can use any other lib / boilerplate** that you might be familiar with as long as you can justify your decision.

- Create React App
- Redux
- Any hosting service to try your app online

# What is important?

- **The app must work**, it is mandatory to have a working example live or at least it should be runnable locally.
- It should be **easy to use**. You should focus on offering a great user experience.
- You should be able to **explain clearly** every decision you took. This includes:
  - Code structure
  - Libs used
  - Why you did not do something
  - Why you displayed something like that and not like that
- Your code should be **simple** and **structured**.

# What is not?

- Trying to cram in as many features as possible.
- Have 100% test coverage

# Stories

Only the first story is mandatory. Other stories **do not need** to be included in order (or included at all): you should do the ones you think will have the biggest impact on the user first.

You should be able to **explain why** you choose to implement the stories in a particular order.

It is better to do only a couple stories with a smooth UX than trying to do everything and sacrifice the experience.

You are not evaluated on the amount of stories you produce but rather on the quality of your work, we know you have a tight schedule. This serves more as inspiration if you have extra time.

## Story 1 (Mandatory)

- I can type the address of my departure and arrival points.
- I see on a map where are the closest stations for my trip.

## Story 2

- I see the closest station from my departure where there is at least a bike, and not just the closest.

- I see the closest station from my arrival where there is at least a spot to park, and not just the closest.

### Story 3

- I see the path to follow between the two stations.
- I see how much time it is going to take me to ride.

### Story 4

- I see how much time it is going to take me to go from my position to the station, and from the station to my destination.

### Story 5

- I can see different path options, and select the one I like the most.

### Story 6

- The app takes advantage of my location in a smart way.