1. Install **Quasar framework** (Quasar CLI) <https://v1.quasar.dev/quasar-cli/installation>
2. Create example SPA (Single Page Application) with name Ride to work by Bike via Quasar CLI, which is based on **Vue.js version 3**, [**Vite**](https://vitejs.dev/), **TypeScript**
3. Init Git repository and use Git much as possible
4. Try to find and integrate appropriate Vue.js version 3 based logging library (logging into the web browser console) and translation/localization library into your Ride to work by Bike

application

1. Your Ride to work by Bike application will use a central config file with the appropriate format (which allow writing comments) with following options TXT, JSON, TOML (choose the right one among them)
2. Ride to work by Bike app config file with appropriate format contain (the values will be used in the your components):

**primaryColor = “#212121”**

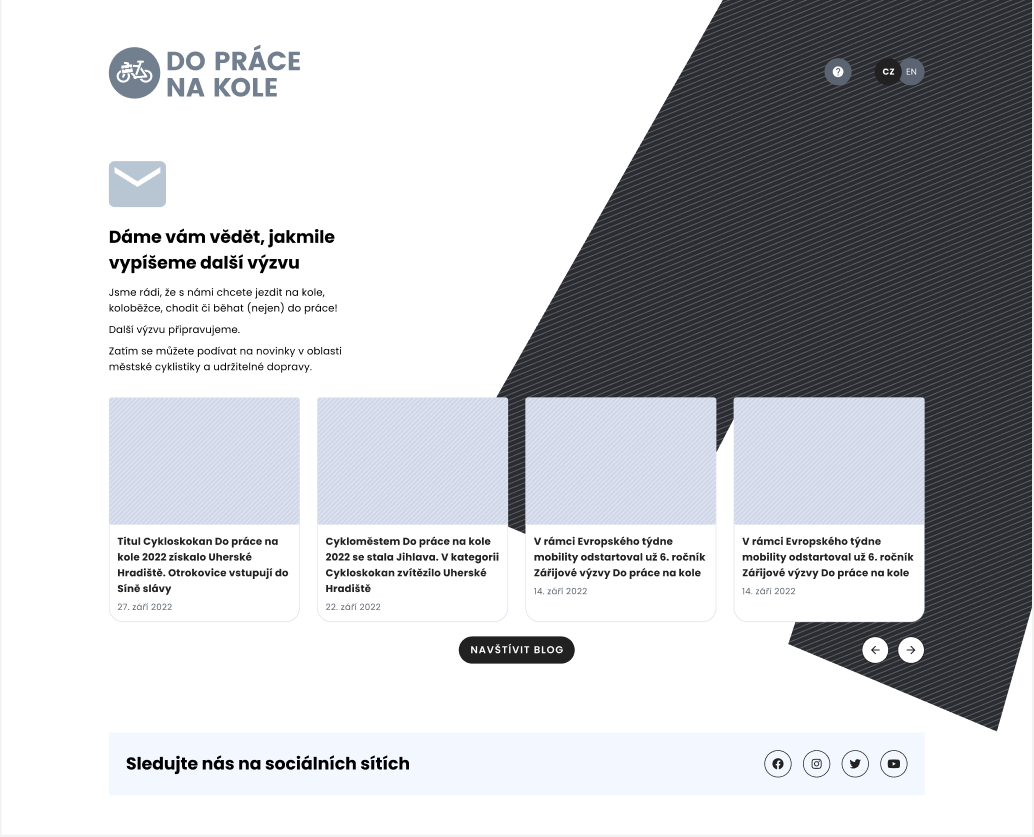
**secondaryColor = “#5b6470”**

**titleImage = “title.png”**

**backgroundImage = “background.png”**

**messageImage = “message.png”**

1. Default Ride to work by Bike app language is English. Translation strings are loaded from the localization files (you need create root directory with appropriate name for these files) with appropriate format with following options TXT, JSON, TOML (choose the right one among them). Try import these localization files into TypeScript code as ES6 module, use dynamic import (require JS Vite plugin content library). Second language is Czech lang. Exact translation is no needed.
2. Create following web page (image bellow), with default route path **/** and accept URL parameter **lang** (example: **/?lang=cs**) for changing web app language. Web page text is not important, but **make them translable**.

**

1. Try to remove from your automatically created Ride to work by Bike app unused and unnecessary code and source code files
2. The application will contain two Vue.js components (image bellow) and choose an appropriate component name and use Composition API for writing components. For components declare some property(ies).
3. **Use Quasar framework UI components much as possible** <https://quasar.dev/vue-components>
4. Override Ride to work by Bike app primary and secondary color with values loaded from central Ride to work by Bike app config file <https://quasar.dev/style/color-palette>
5. Change app language via buttons widget (image above) or via default route URL parameter **/?lang=cs** (check the point no. 9)
6. Log some information from components into web browser dev console (check the point no. 4)
7. Try integrate [Cypress testing framework](https://docs.cypress.io/guides/overview/why-cypress) into your example Ride to work by Bike application <https://github.com/quasarframework/quasar-testing/tree/dev/packages/e2e-cypress>
8. Be familiar with [Cypress testing framework](https://docs.cypress.io/guides/overview/why-cypress) and with options how to run tests
9. Try to write [component test](https://docs.cypress.io/guides/component-testing/testing-vue#Selecting-the-Stepper-Component)  for every component and one [end to end test](https://docs.cypress.io/guides/end-to-end-testing/writing-your-first-end-to-end-test) for app route path / and if you need write unit test with [Cypress testing framework](https://docs.cypress.io/guides/overview/why-cypress)
10. In component test try to test if images e.g. titles.png, background.png, message.png are loaded properly
11. Run component/e2e tests with different web browser Mozilla Firefox, Chromium, Google Chome/Microsoft Edge
12. Try to change 404 page text and make them translable
13. Try to build Ride to work by Bike app for the production