CSCI 362-02-2020: Chapter 1/ Deliverable #1

Team 7-11: Justin, Matt, and Janneke

Our project choice

After attempting to compile several projects, we finally chose our top candidate, **wheelmap-frontend**[1]. Wheelmap is a crowdsourced online map to search, find, and mark wheelchair-accessible places. Wheelmap-frontend is the newer, maintained repository containing the Node.js/React.js-based frontend for the Wheelmap app.

Compiling the project

A quick note about the virtual machine:

I (Janneke) had never used a virtual machine, so through compiling this project I learned to set one up so that I could run the project on Linux. I used VirtualBox/Ubuntu. The process was fairly straightforward using a Medium article; however, I later faced some issues with the virtual machine failing to fire up. Since I haven't taken CSCI 340 yet, troubleshooting these issues was a valuable learning experience.

Fortunately, we faced relatively simple issues while compiling this project. The following table compares the repository's instructions for compiling wheelmap-frontend against our actual sequence of terminal commands:

Build instructions	Actual build sequence

`# Environment variables`
`cp .env.example .env`
`# npm dependencies`
	`# clone the repository`
	`git clone`
	`https://github.com/sozialhelden`

[1] https://github.com/sozialhelden/wheelmap-frontend

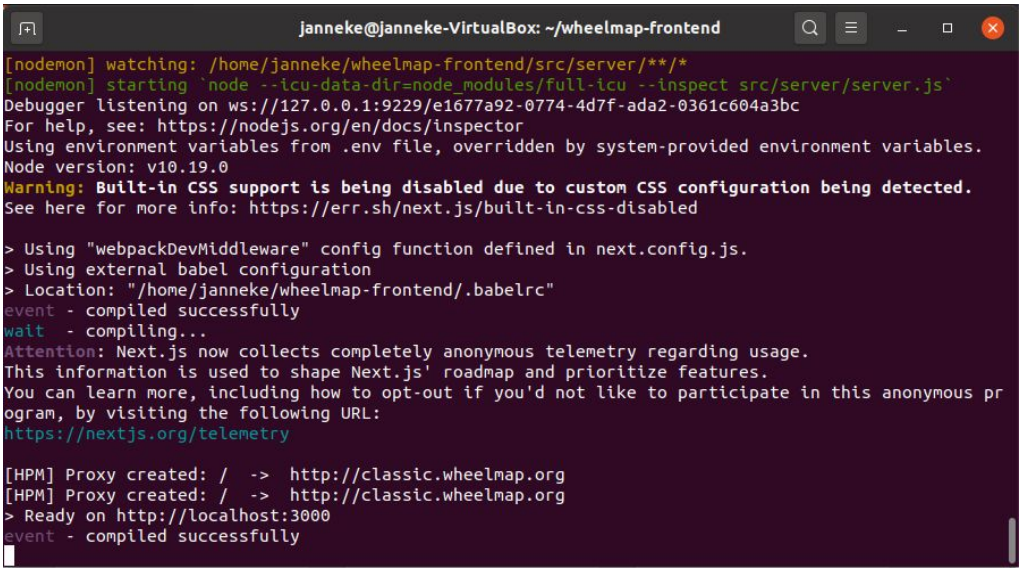
npm install # install transifex i18n / localization tool pip install transifex-client # start a local test web server npm run dev	/wheelmap-frontend # enter the cloned repository cd wheelmap-frontend # environment variables cp .env.example .env sudo apt install npm sudo apt install transifex npm run dev npm rebuild npm run dev

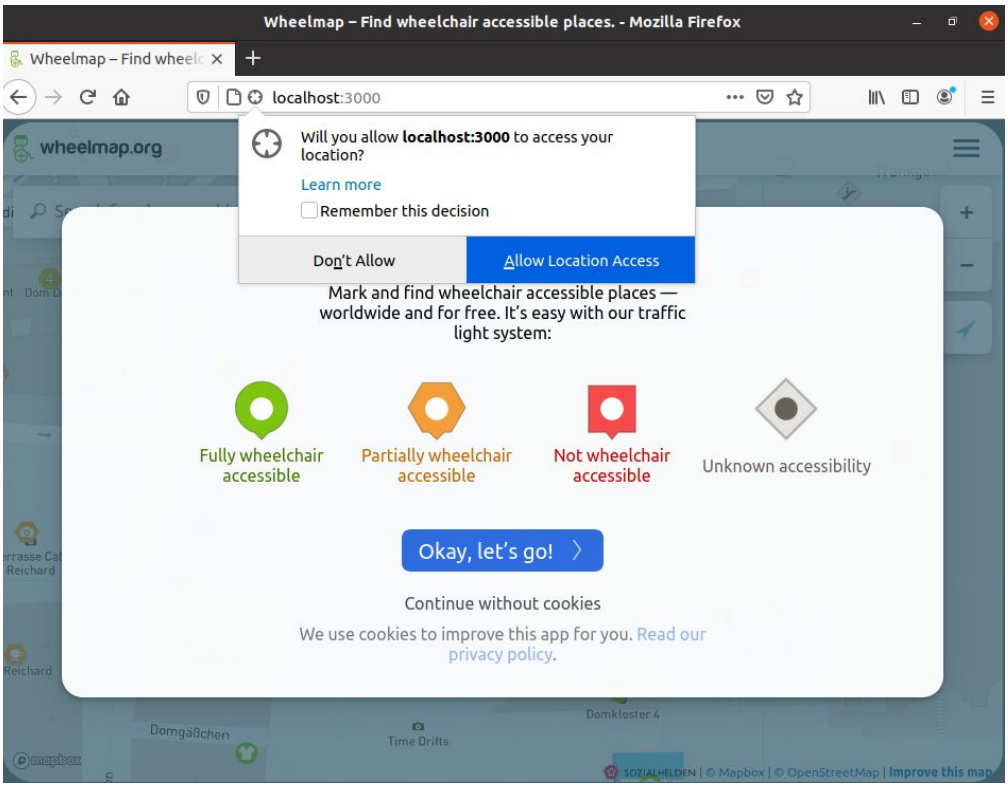
One of the biggest pain points was installing Node Package Manager (npm). This is the default package manager for the JavaScript environment node.js. After running `npm install` per the directions, we ran into the error repeatedly: "`npm WARN Local package.json exists, but node_modules missing, did you mean to in install?`"

Running `npm rebuild` resolved the issue. After looking into the documentation[2] for npm, I found that rerunning `npm install` or running `npm rebuild` will compare the existing "node_modules" installed by npm to those in the most current version of npm, only installing what is missing. We are still looking into why this function would resolve the issue immediately after we installed npm and, presumably, the most up-to-date packages.

After the npm issue had been resolved, we received the message: "Ready on https://localhost:3000" and were able to open a local version of Wheelmap on Firefox!

[2] https://docs.npmjs.com/cli/install#algorithm



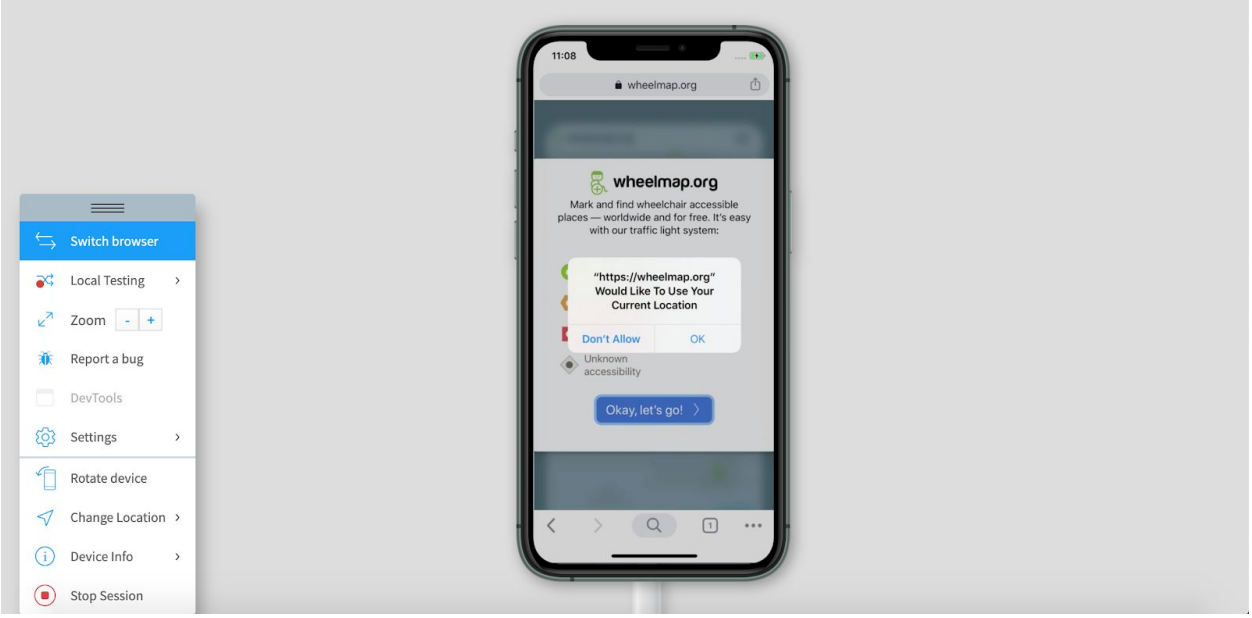


Testing

The wheelmap-frontend repository README.md page has the following note about testing: "For testing the apps, we use BrowserStack - it can run test suites on various browsers and live devices. Currently, our testing happens mostly manually on BrowserStack Live, but pull requests

will soon get automatic CI checks using BrowserStack Automate and BrowserStack App Automate. We thank the BrowserStack team for their great products and their support!"

We made free accounts on BrowserStack to investigate and found that it provides live testing (limited in the free version) which allows you to select a device/operating system/browser, then test a series of steps. You can also automate this testing through BrowserStack Automated. However, since we are not affiliated with Wheelmap and do not have the paid version, we were unable to access any tests they have automated and organized through the application.



Within the "docs" folder of the repository, we found a testing document[3] which outlines an extensive series of steps to test the main functionality of the wheelmap react frontend application, which are performed by manually clicking around the application. Here are just a few examples - there are 28 sections like the ones below:

First time usage

- 'welcome' dialog shown on first start
- 'welcome' dialog can be dismissed with the 'lets go' button

[3] https://github.com/sozialhelden/wheelmap-frontend/blob/master/docs/TESTING.md

- tapping any button or the blurred does not trigger any reaction on other underlying components
- 'welcome' dialog can be re-opened by tapping the logo in the main menu
- Browser(tab) title matches application name

Positioning

- on first start the user is asked for location access

Behind the scenes, the Wheelmap team likely has these tests scripted and automated within the BrowserStack app, though it seems they have not publicized any test code. We are working on reaching out to the Wheelmap test to see if it's possible to earn access to the testing scripts.

Evaluation of experience

So far, this project has proved a great learning experience for each of us. We got to interact with several different projects, evaluating their strengths, weaknesses, and deal-breakers before finally choosing Wheelmap. This sheds light on the importance of thorough documentation because in all cases, that was the deal-breaker for us - we ran into errors attempting to compile Cadasta and Martus that were not worth pushing through when our other candidate, Wheelmap, was so clearly documented. Our next steps include learning more about the existing testing process for Wheelmap looking ahead to the second deliverable.