Kirill Kirillov - 62648727 Gavin McGill - 59048032 Luke Woollett - 47870172

INFO 263 Assignment 2017 - Group O

Auckland Transport Real Time Bus Plotting

Website functionality

The website's functionality could be split into three main parts. The first being loading in all the available routes by pulling them from the akl-transport database and then populating the selection box with them. Secondly it would query the Auckland Transport API for a route which was selected by the user. Finally when the data was received by the browser the google map was updated to show the busses that were currently present on that route or the user was informed that there were no busses on that route.

The first part involved a jQuery document ready function to send a post request to the server. This would then pull all the routes from the database and send that back to the browser. We then used Javascript to populate the selection box with all the routes. This was needed to be achieved client side as it was much easier and more readable to implement using Javascript instead of echoing out HTML with PHP.

The second part involved querying the database with the given route name to extract all the tripids. A jQuery POST request was sent to the server with the selected route in order to extract the tripids for that particular route. The tripids would then be used to query the Auckland Transport API.
After the tripids were sent to the Auckland Transport API and JSON containing the bus locations and other information was returned. The relevant information then had to be extracted (latitude, longitude, busid and start time). This was processed on the server-side to make the client-side faster to load and an overall better experience for the user. The server would return nicely formatted JSON which was extracted by the browser and the bus locations were plotted onto the map.

Group reflection

Overall the group functioned rather well and tasks were completed even though we only had 3 people. We found that having a smaller number of people was much easier to have good communication and to allocate people to various tasks. We had enough trouble trying to get people to work on various things at the same time due to how small the project was. We would sometimes find that we couldn’t all do our own work because we would have to have people working on the same file which would introduce merge errors with Git.

Our main method of communication was through facebook messenger and also face to face as we all shared the same classes. Messenger was primarily used to remind people or to send quick messages for things that needed to be done that didn't require a great deal of explanation, whereas a face to face meeting would be used for more complex tasks that needed some more in-depth explanation.

Our code was managed via Git, this was quite good for various reasons, the main one being that we were all rather familiar with it. We only had a few issues when someone would use 'git add --all' and would add all the junk files which would cause merge errors. This was later fixed when we all started adding only the files we worked on instead of everything. Git was also helpful for looking back at previous code to see how it was changed in case anything did go wrong so that we could revert or alter any of the affected code.

We separated the tasks into two main categories, the front-end and the back-end. This allowed us to simultaneously work on different things without the fear of merge errors occurring if someone was editing a file that someone else was at the same time. We would break the five problems into smaller chunks and work on them in various order to ensure we were being productive the whole time. For example when the API query wasn't working too well we would have only one person trying to figure it out while the rest of the group would work on changing css or completing one of the other tasks. This allowed all of our group members to work on something so that we would not delay the completion of the project. In some circumstances two people would pair program if there was something a bit more complex in order to be able to bounce ideas of one another and to complete the task more efficiently.

Reflection

The things that did go well was that we all had fairly good communication with one another most of the time and we didn't have too many merge errors or problems that had us stuck for a long time. We also delegated work so that people could work simultaneously.

The things that did not go so well was that when we did have miscommunication it ended up being that quite a large chunk of code was re-written because of this. Another problem occurred with Git and us adding everything, including the junk files which made fixing merge errors quite frustrating.

If we were to do this assignment again we would spend a little bit more time planning how we would approach the tasks as we underestimated how long each task was going to take and would distribute the workload a bit more towards the back-end instead of the front-end in as that was where we struggled with the most. We would also figure out how to use an IDE such as phpstorm because expression web isn't very good. It lacks basic things such as bracket matching and highlighting all variable names that are the same as the selected on in the document.