Final Report: Team Fresh Air

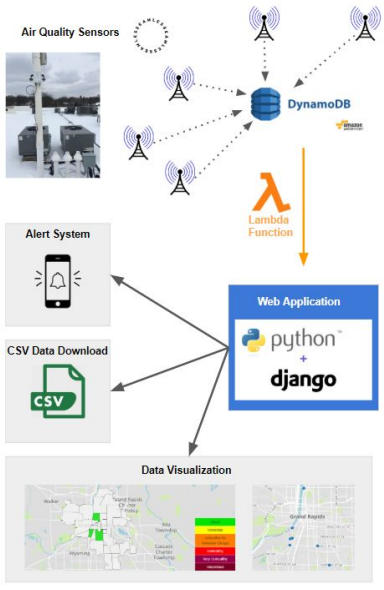
Prepared By: Tressa Groelsm, Javier Ramirez, Chris Schertenlieb, Megan Thomas

Abstract:

Awareness of the environments effect on our health is growing daily. Concerns with air quality has been one of the most rapidly growing environmental concerns. The average person inhales 14,000 litres of air every day. If this air contains any pollutants it can have both long and short term effects of your health. People with lung conditions such as COPD (Chronic Obstructive Pulmonary Disease) and asthma are significantly more susceptible to health issues related to poor air quality. While air quality is a constant effect on our personal health, accessing information on the air around you is not as easy as you would think. This issue led us to the development of our senior capstone project. The goal of our project was to create a community-focused, educational, web application designed to inform Grand Rapids residents of air quality metrics. Our web app uses data gathered from sensors around the city to visualize, interpret, and manage pollutant information. Along with displaying metrics, we provide citizens with the ability to download open source air quality data as a CSV file for other projects. Our project includes a community outreach program that sends emails and texts to registrants about relevant air quality conditions in the city. To promote inclusivity our web app is available in both english and spanish. The goal of our project was to provide this data in an easy to use manner that is accessible to all.

Introduction:

To bring our project to fruition we had implement various technologies and skills. For ease of understanding we created a diagram that demonstrates the basic flow of our project. As depicted to the right, there are several air quality sensors placed strategically around the city of Grand Rapids by our sponsors. The sensors collect and store the data locally until network connection is achieved. Onced the connection is reached the data is then sent to an Amazon DynamoDB. We then used the Amazon API to retrieve data from the database and bring it to our web app. From the web app there were three different uses for the data. First of which was checking air quality data with current standards to see if an air quality alert is needed. The homepage of our web app prompts users with a sheet to sign up for text alerts. There you can choose



your level of alert as well as your preferred method of contact. Next was using python libraries to create data visualizations for the web page. There were two main areas that we displayed our data, a map to show sensor locations and a heat map that displays the pollution levels in each area of the city. Lastly we allow users to download air quality data as a CSV file. We have a tab dedicated to community access to the air quality data we collect. This data is downloaded simply by clicking one button. We were also aware of the large hispanic community in Grand Rapids so we created a translated version of the website to cater to that demographic. We were also aware of the lack of knowledge surrounding the effect of poor air quality. This led us to research and create an easy to follow guide focusing on the effects of each pollutant. We also created an About page for the community to learn more about the project and any future projects created.

Team Fresh Air:

Web Development

The framework we used to develop our web application was Django.To meet sponsor requirements we needed to include several pages. The home page presents users with a form to sign up for air quality alerts. The About page has several pages for the sponsor to update with information about the project as it progresses. This page also contains a map with all the air quality sensors locations in the city. The Air Quality Report gives the current air quality metrics and presents them as a heat map laid over the city. The Air Quality Guide page contains information that we researched to inform users on the importance of air quality. We also have a Get The Data page for citizens to download and use the data we collect. Lastly we have a Contact Us page for people to submit questions and issues.

URL Configurations

We are aware that as the project grows that additional changes may need to be created for the web application by the programmers this is handed to. For their knowledge we are including our URL configurations for better understanding and ease of use. We also have this information provided with the script documentation. The ‘urlpatterns’ list routes URLs to views. Here we provide several examples for clarification:

Function views

1. Add an import: from my_app import views
2. Add a URL to urlpatterns: path('', views.home, name='home')

Class-based views

1. Add an import: from other_app.views import Home
2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')

Including another URLconf

1. Import the include() function: from django.urls import include, path
2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))

For further general information please see
https://docs.djangoproject.com/en/2.1/topics/http/urls/

Data Management
The raw data that we received from the air quality sensors is sent to a Dynamo database. From here we pulled the data and cleaned it using several noise reduction techniques. The different pollutants pulled from the sensors are all analyzed and the different risk levels are compared. This data was then used for data visualizations, the alert system, and the downloadable CSV file. We were also expecting more sensors to be up and collecting data by this point but due to a variety of circumstances this was not possible. This is why we designed our data management section to easily adjust as sensors are added into the locations table.

Database Structure
The dynamoDB database that Seamless provided had three tables, one of which is currently not being used. Rather than creating our own database to store our own data we simply created tables in the existing database. Over the course of the project the structure of the database changed and this is now the current structure:

email_list(email, carrier, emailAlerts, phone, textAlerts, teir)

hourly_deq_data(date, ozone, pm25, wind_direction_degrees, wind_speed_mps)

mapdata(time, sensorAQI, sensorIDs,)

ost_data(dev_id, time, o3, pm10Average, pm1Average, pm25, pm25Average, pm25Count, pm25Max, pm25Min, pmStatus, port, specHumid, specO3Count, specO3max, specO3min, specStatus, specTemp)

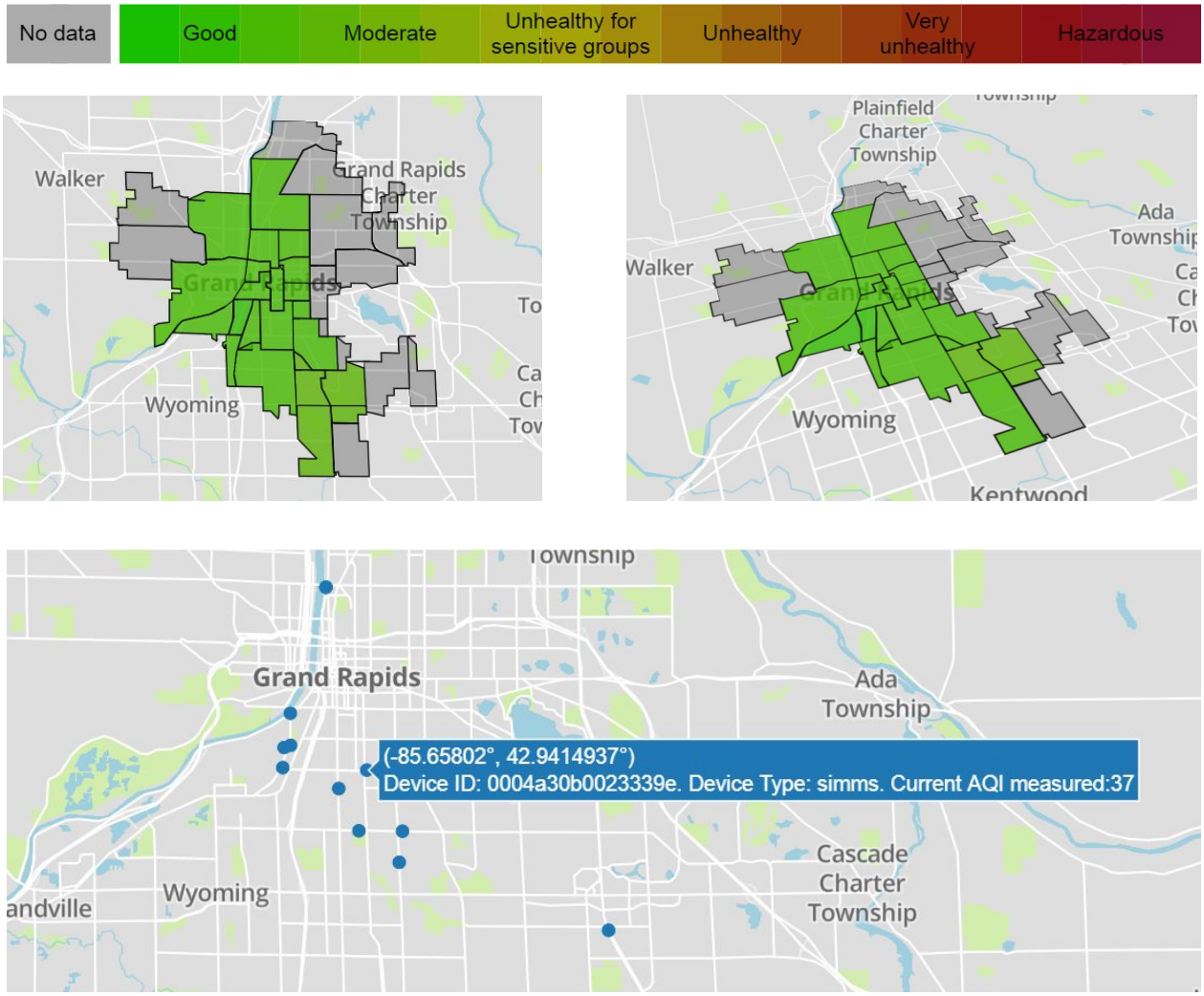
sensor_locations(device_id, device_type, device_gps_location)

simms_data(dev_id, time, co, humidit, no2, no2_o3, o3, pm25, so2, status, temperature, vin)

Data Visualization
Having a way of presenting the available data in a format that would be quick and easy to understand for all users was a priority for the team. We decided to go with two map representations of the data, for which we used the Dash library by Plotly. This library was extremely helpful and it aided our development process greatly, helping us create two interactive maps that have functionality such a zooming, 3D graphics, responsive object hovering and other features.

The first map simply consists of a representation of all the air quality sensors that Seamless has set up in Grand Rapids. If a user hovers with their mouse over the points, it will display the name of the sensor along with some other information, such as the currently measured Air Quality Index. The second map is a heatmap that indicates the

different AQIs measured throughout the different neighborhoods of the city. In order to calculate the AQI of a certain neighborhood, all the measurements from the different sensors in that area are averaged. We also created an algorithm that estimates the AQI in zones were there is no data by using the data from the adjacent areas.



Alert System

We wanted to give the community the ability to monitor the quality of the air around them. Our alert system allows the community to have alerts about poor air quality at a moments notice. We have a sign up page that allows user to sign up for either alerts via text message or email. The scripts we created used the smtplib python library. This library was specifically designed for sending emails, which made running that section of the alert system fairly straight forward. The text message script was not as simple. To keep our script from using any paid third party platforms we decided to ask users for their phone carrier. This allowed us to send text messages from our email account entirely within the script. The account information for sending the alerts and accessing

the database are stored in a separate secure configuration file.

Areas of Technical Growth

Each of us experienced tremendous growth over the semester. This growth was in a variety of areas from business to social. However, our largest area of growth was technical. Being that we call came from different technical backgrounds we each had varying levels and areas of growth.

> Megan: While I took web apps I didn't get this level of exposure to web development. I had no experience with bringing this type of data through a web application. Beyond that I had no experience designing the architecture of a project on this scale. This project helped expand and make concrete my web development skills. The second technical skill I feel I developed was my experience with Python. The computer science program here starts us with Java but as I continued with the project I used C fairly heavily. This project gave me an exposure to Python that hadn't seen previously.
>
> Chris: Going into the project I was the most familiar with Web Development so I took the lead on that. While I didn't grow much in that respect, I learned a lot of things about Python and Django. Using Python for web development was very different from any other kind of framework I've used in the past. I enjoyed mixing procedural programming languages with web development technologies. I also learned a lot about Python development in general, I learned how to (properly) use pip, I learned how to set up a virtual environment, and I learned a lot of neat Python tricks I haven't been exposed to. I also learned a great deal about AWS from this project, something I had previously only used once. I now feel familiar with DynamoDB and Elastic Beanstalk.
>
> Javi: Previous to this, I had never had to work a project of this scale and that involved so many different parties. In this aspect, working on this project has greatly improved my organizational skills when it comes to putting together a plan for the development and completion of a big project, which will most definitely be of use in the future. I also improved on my ability to communicate, both with my team and our corporate sponsor. On the more technical side, having to store, manage and compute large amounts of data is something that I was always interested in doing and I got exposed to in this project. Having to use Python, along with some external libraries has also been good for me, as I've had to learn and adapt to a new way of coding.

Tressa:

Software Engineering Code of Ethics and Professional Practice:

In our initial prospectus report we selected 8 ethical principles that we would strive to uphold. Here we will discuss how we actively worked towards each of these principles over the course of our project.

1.01. Accept full responsibility for their own work

Responsibility for work was handled in our personal journals as well as our GitHub commits. We felt this did not entirely encapsulate the amount of work we had each done so we made it a priority to meet twice a sprint. We started the meetings by each discussing our personal progress, or occasionally lack thereof for extenuating circumstances. This helped our development to ensure we were all contributing to the project in a way that the group was happy with. We also meet with our sponsors once a sprint to ensure they were pleased with our individual progress.

2.01. Provide service in their areas of competence, being honest and forthright about any limitations of their experience and education.

We worked to uphold this principle starting from our initial meeting. In our first meeting we discussed the classes taken, relevant person projects, and any expertise we had. This was then used to decide how we would divide our project and how we would delegate tasks to each member. We then followed these tasks for the entire project.

3.01. Strive for high quality, acceptable cost and a reasonable schedule, ensuring significant tradeoffs are clear to and accepted by the employer and the client, and are available for consideration by the user and the public.

By maintaining the previous two principles we consequently worked towards principle 3.01 as well. By being forthright about our abilities we could ensure that we had the best developer for each of the tasks that we needed to complete, maintaining the highest quality. Meeting with our sponsor at the beginning of each sprint created a transparency between us which we used to discuss our schedule and progress.

5.12. Not punish anyone for expressing ethical concerns about a project.

A goal of our project was to provide air quality data to **everyone** in the city of Grand Rapids. To provide the best possible product there were many ethical issues that needed to be considered when creating an application for the general public. We encouraged ethical concerns to be raised so that we could better cater to our users.

6.08. Take responsibility for detecting, correcting, and reporting errors in software and associated documents on which they work.

Our weekly meetings provided plenty of opportunities to review code and documentation. These meetings also created a closeness within the group where no one felt uncomfortable taking responsibility for errors if they were found.

7.02. Assist colleagues in professional development.

When delegating tasks we realized that not everyone is an complete expert in area at this stage in our careers. To assist with that we often used slack and weekly meetings to help each other. Also Megan's primary goal was the alert system but she was also delegated as a full stack developer. Her "full stack developer" position was designed to help any other members work on issues if they were overwhelmed or needed assistance.

7.03. Credit fully the work of others and refrain from taking undue credit.

We worked towards this in our journals, GitHub commits, and weekly meetings. This provided us with multiple documented sources on who worked on which aspects of our project.

8.01. Further their knowledge of developments in the analysis, specification, design, development, maintenance and testing of software and related documents, together with the management of the development process.

Being involved with every step of the development process was a learning experience for each of us. We all had different degrees and areas of technical experience so we learned from each other and from our sponsors. This project exposed us to new technical and business based challenges. Biweekly meetings with our "boss" gave us more business experience that we have only seen in internships thus far.

Teamwork Reflection:
Team projects are a staple of the computer science program here at Grand Valley. Our years of previous experience had helped to teach us how to be effective in this group setting and that was reflected throughout the project. While we all had our individual tasks, we often met and worked together. There was no hesitation to reach out if anyone needed assistance. While this teamwork was instrumental for our success we each had our individual take on the project:

> Megan: Our team worked extremely well together. We each were active in the group chat, came to all meetings, and contributed the best we could. There isn't a single member on the team that I think put less in than the rest of us.
>
> Chris: Hands down the best group I've had for any assignment in my entire CS career. I can't think of a single complaint about any of the group members. Everyone worked hard and communicated well. We had great chemistry and everyone was very responsive and helpful.
>
> Javi: I feel really lucky to have had such a good team for this project. Every single member has been proactive on their work and shown initiative and excellent technical skills since day one. Communication was fast and effective thanks to our group chat, which made it possible to work separately for most of the project. Group meetings were very productive as everyone was prepared for them, and they helped us set realistic goals for each sprint.
>
> Tressa: This group was a very easy one to work with, and it was honestly a breath of fresh air (pun intended) to get to work with a group that communicated this well. It was easy to combine our work, group meetings were incredibly productive, and I would be more than willing to work with any of these people again in the future.

Conclusion:
As our project came to a close we were satisfied by the work we completed this semester. The non-trivial features initially laid by our sponsor were management and analyzation of sensor collected air-quality data, a web application, email notifications, and updates feed for dashboard display in clinics. Implementing these features was a priority to the team, as they provided the basic functionality we believed the project should have. The only one of those that remained uncompleted was the dashboard displays. This was because we could not get access to the technology available at each

clinic and this was eventually dropped from our list of tasks by our sponsor. We also went on to complete several of the supplemental features including text alerts, spanish translation of website, CSV data downloads for citizen scientists, and educational section with guides on air-quality effects. There were several challenges that we faced along the way but we were able to overcome and be successful.

Challenges and Successes

We are all full time students with at least one job. We are all seniors taking some of our most difficult course and dealing with major life changes. Our biggest issue was having enough time and energy to commit to the project. Our busy schedules made group meetings a bit of a pain. Luckily our communication skills helped us the most here. Being available on slack for help, being flexible in group meetings, and expressing concern when it was needed allowed us to complete as much as we did during this project.

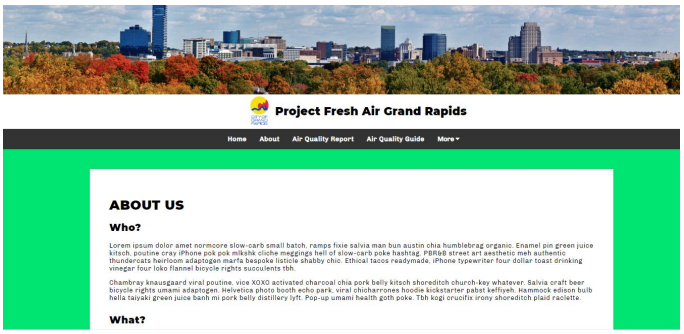
Our course structure was slightly disrupted this semester as well. We lost an entire week of classes due to inclement weather. While this didn't disrupt a lot of class time it did impede our work. It was difficult to leave the house to go to campus to work and some of us were left without power. Beyond that there was a sprint that consisted of midterms week and spring break. While this threw off our original sprint schedule we worked as a collective to make up our previous lost time.

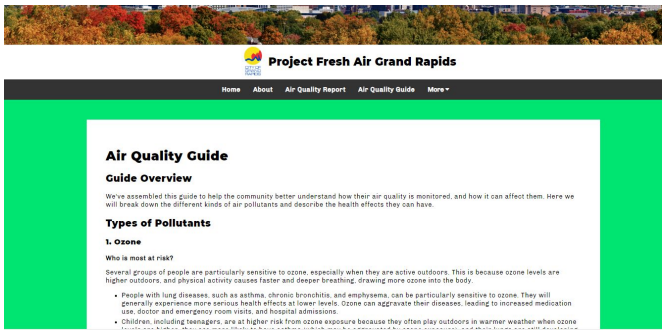
Overall the computer science department was instrumental in teaching us skills to be successful here. The course CIS 350 software engineering helped us use the agile style of development and taught us how to develop many of the documents needed throughout the course. Several of us took web application development which was particularly helpful in planning and developing our specific project. Some members of our team also had experience in data handling and cleaning which was also very helpful. Each member of the team brought a skill to our project that led to our success.

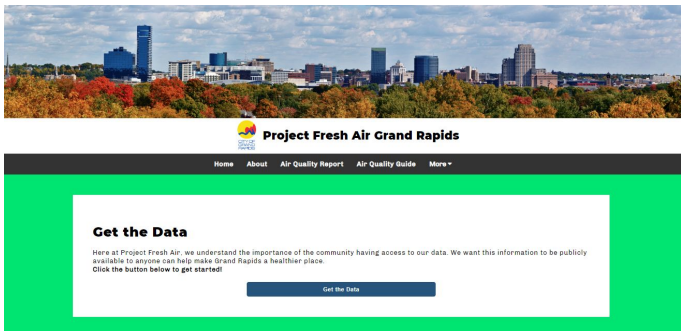
Future Work

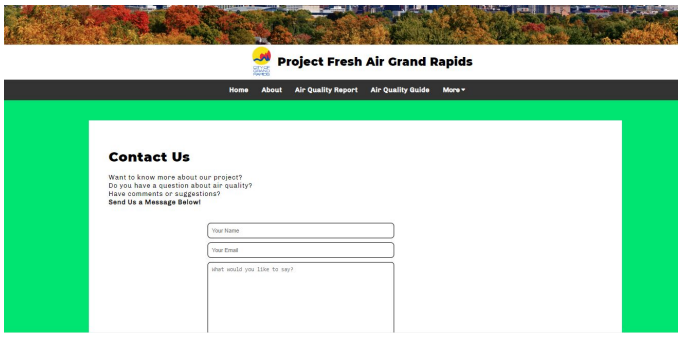
Our web app is just the first step towards better air awareness. We added a page to our app that allows citizens to download air quality information for themselves. This data is open source and free to use, provided by the Seamless IOT team. We hope that this data can be compiled and save for citizens personal projects or possibly capstone groups in the future to access. Our group, partnered with Mercy Health, are working to compile air quality data and HIPAA safe information from Mercy Health's clinics for future research on how air quality affects those with pre-existing respiratory issues. Hopefully the combination of these data sources can lead to important future work.

Screen Shots

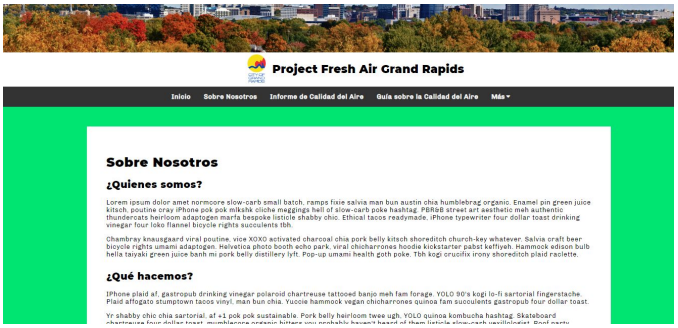


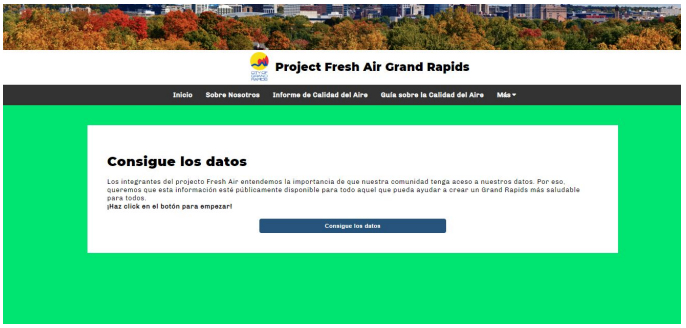


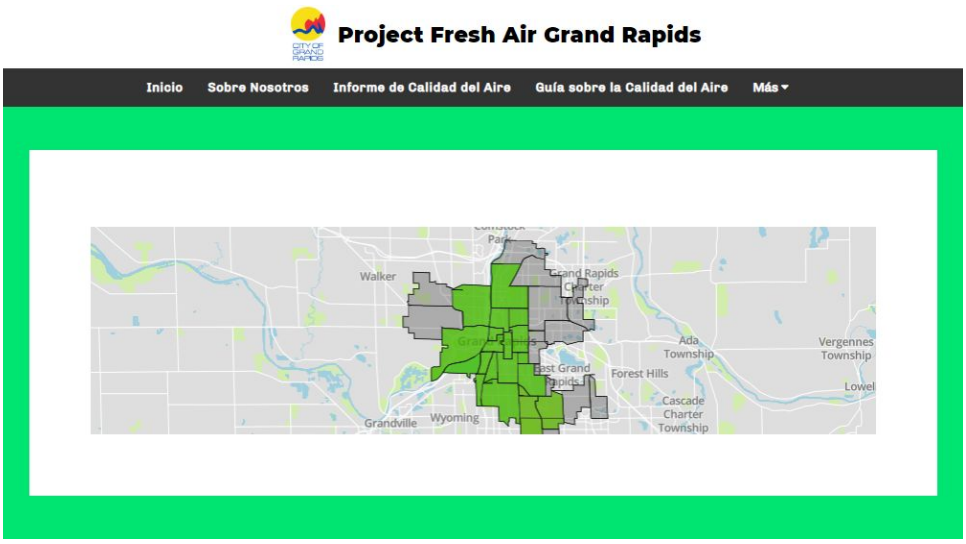




Screenshots of spanish translated site







Responsive Screenshots