New Jersey

Carbon Dividend Calculator

Final Report

Jonathan Lu – Project Leader

Agata Foryciarz

Changyan Wang

Joseph Abbate

New Jersey Carbon Dividend Calculator

History

Ideation

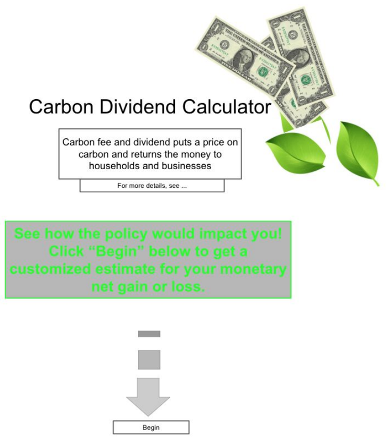
We met in the dining hall to brainstorm over a meal in January. We first thought to make a note-taking app similar to WorkFlowy or Evernote that would send reminders like Boomerang does, but we quickly realized this would be extremely low impact. Energized by the thought of making real impact in the world, we considered instead doing “something about Trenton”. Unfortunately, this lacked concreteness. We then moved on to the idea of giving an interface for constituents to directly contact policymakers in a social media-like site. Unfortunately, this idea was already filled by Countable. Ultimately, we left our lunch meeting without an idea.

Nonetheless, over the following month we kept a background process of ideation going in our minds. Ultimately Jonathan, through his work with the Princeton Student Climate Initiative, realized there was a real need for a calculator to go hand in hand with the climate policy they are proposing. This is extremely impactful: first because it gives citizens direct access to the way a policy will specifically affect them, and second because it can help advertise an action that will directly help mitigate climate change. It is also very concrete since Jonathan already had plenty of background knowledge on the policy and the calculations we needed to do.

Initial Planning

We met with Professor Kernighan very early in this process to present our idea in powerpoint format. Our original design laid the basic functionality, but left many details to be fleshed out. Nonetheless, we accepted that our idea was imperfect and moved straight into the implementation as we began meeting with professors, activists, and students to get feedback.

We met every week for at least 4-5 hours total with the initial goal of getting a Django framework working with two basic views (presented at the top of the next page), and all running on Heroku with a PostgreSQL database. This took longer than expected: it took us 3 weeks to get the basic website working. This initial barrier was one of the hardest we faced. Still, after getting the basics working, we could start dividing up the work and making tangible progress at each meeting.

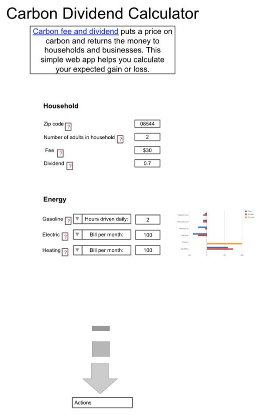


Carbon Dividend Calculator

Carbon fee and dividend puts a price on carbon and returns the money to households and businesses

For more details, see ...

See how the policy would impact you! Click "Begin" below to get a customized estimate for your monetary net gain or loss.



Carbon Dividend Calculator

Carbon fee and dividend puts a price on carbon and returns the money to households and businesses. This simple web app helps you calculate your expected gain or loss.

Household

Zip code ?	08544
Number of adults in household ?	2
Fee ?	$30
Dividend ?	0.7

Energy

Gasoline ?	Hours driven daily:	2
Electric ?	Bill per month:	100
Heating ?	Bill per month:	100

Milestones

We began by evaluating other carbon calculators available online. We found that simplicity and transparency of the app were of utmost importance. We then began sending emails to researchers who implemented these calculators and to Princeton professors to get initial guidance on our implementation. We maintained momentum on this front throughout the project, continuing to absorb and adjust to the knowledge we gained from these discussions (see User Research).

We set very ambitious timelines for ourselves, and generally stuck to them better than we had expected. We aimed to have a working prototype by the end of spring break despite having no web development experience, and we did deliver on this. We also planned to have our beta test ready by April 18; thanks to lots of meetings leading up to the date we also delivered on this.

Nonetheless, in the beginning of the project we did set overly ambitious deadlines for getting the piping working: our first self-deadline was to get the Django framework up on Heroku within the first two weeks, and that ultimately took until Spring break. Also, we planned to have the final report written by April 30, and it's currently May 10.

Surprises

On a high level, we were surprised by how easy seemingly hard things were, and how hard seemingly easy things were. In particular, getting the basic plumbing working is in theory a very simple task, but it ended up taking us 3 weeks to fully complete. Yet the transition from our very simple website to a beautiful user interface, Bootstrap made very smooth. When we went on to implement the simple feature of sticking the graphs on the actions page to the top of the browser window as the user scrolls, we imagined that there would be a simple JavaScript library (e.g. from JQuery) that would take care of everything for us. However, the implementation of this feature turned out to be highly non trivial for our specific situation. In sum, the lesson learned is that how "cool" a website looks isn't necessarily a good proxy for how hard or easy it was to implement.

Heroku's file system is ephemeral. Effectively, the only files stored permanently are those which are tracked in the project's Github. This was a problem for our Django settings. Because settings.py must be different on each computer, we had planned to untrack settings.py from Github, and send a custom settings.py directly to Heroku. However, this settings.py would be lost due to the ephemeral file system. In the end, we settled on a fix where each team member kept uncommitted changes in their personal settings.py file, and avoided pushing changes to settings.py unless these changes were needed on Heroku.

Product Future

One technical stretch feature we were unable to execute is updating average values based on household size and zip code (e.g. a 10 person home likely uses more energy than the average 2.73-adult NJ home). Another is to allow multiple different mechanisms of inputting data (e.g. giving heating bill per month rather than number of therms, mpg and miles rather than just gallons of gasoline, etc.). For this idea, we would hide input types under radio buttons to avoid visual clutter from too many input types. AJAX would likely be useful in implementing both of these stretch features.

On the nontechnical side, the primary goal moving forward is to share this work with assemblymen and activists to allow it to start helping both constituents and legislators in the real world. As the Princeton Student Climate Initiative continues developing the policy and working with groups from around the State and the Eastern seaboard, we will continue to receive feedback on our app so that it might ultimately be publicized to all NJ residents, and perhaps be adapted to work for other states (a relatively easy technical endeavor).

Team Organization

We consistently met as a full group 2 to 3 times per week for two hours, and got most of our work done during that time. We would discuss progress at the beginning of each meeting and decide on tasks for that day's session. Working in the same space allowed us to quickly resolve difficulties collectively. It also allowed us to ensure in real time that our changes were mutually compatible.

One useful device for us was a single group log (a Google doc we all had access to) on which we

1. Took notes on user feedback and meetings with professors / activists
2. Maintained a list of to-dos at the top, which we could update throughout work sessions
3. Took notes on major accomplishments from each work session to track our progress

From a technical standpoint, each of us quickly gravitated towards a different part of the implementation. Jonathan, who had previous experience with Django, handled the Django implementation. Changyan quickly became the database and Heroku expert of the group. Agata concentrated on the front end bootstrap implementation and made design decisions. Joe took charge of implementing the JavaScript-based interactive features and the implementation of cookies.

User Research

Our app is relatively simple on the backend, but it is part of an extremely high stakes movement to help mitigate global warming and therefore needs to look just right from the front-end to be useful to users. The most important aspect of our work was therefore getting many iterations of user feedback. Here we describe key encounters with users (both on our PowerPoint-based prototype and the real deal later on) and the primary lessons taken from each.

Elke Weber 1 (2/7)
Elke is an expert in the psychology of climate change communication. She mentioned that we should avoid using “negative” words like tax as much as possible, which ultimately guided us, for example, to call our project the Carbon “Dividend” rather than “Fee and Dividend” Calculator. She also mentioned that what people really want is a feeling of control: that they will be able to “do something” to offset the cost increase should they choose to do so. This guided us to create our “Actions” page.

Alan Borning (3/5)
Alan, a professor at UW, created the “Carbon Tax Swap Calculator” a few years ago, which had a similar vision to our own but excluded actions and had a more in-depth inputs section. One big lesson from him was that “people like sliders”: we originally were going to have users type numerical values on the actions page, but ultimately decided to have them input these values with JQuery sliders. Another key lesson was that objectivity should be central to our implementation. This guided us to exclude our originally planned logo (leaves and money) and to take out any language that promoted the policy as a mechanism for reducing climate change. We had a turning point, which informed the rest of our implementation and subsequent discussions, that this calculator would be all about the user’s “bottom line” (it’s all about the money).

Chris Jones (3/8)
Chris is with CoolClimate, a Berkeley-based research group. His site is an interface for calculating one’s carbon footprint and looking at ways different actions can reduce that footprint. We noticed that the page was far too cluttered, and this incited us to limit our number of actions to the three most important ones we could find, with one for each area (gasoline, electric, and heating). Most importantly, we received an incredibly rich dataset out of this meeting, which details about energy expenditure across the US. This dataset is the data we plan to use for calculating the customized defaults for each zip code.

ACM 1 (4/6)
This was our first test of the actual website on users (up to now we had been using our PowerPoint prototype). One of the greatest complaints was simply about the text describing the policy: people needed examples to understand the way the calculation is done, they couldn't just be described the process abstractly and understand it. People were also deeply offended by our use of technical terms, like "10% ethanol" rather than "regular gasoline". Finally, people noted that on our actions page it was dumb that the cost breakdown at the top failed to scroll down with the user, so that whatever changes were being made by the sliders were going unseen outside of the browser window.

Ted Borer (4/17)
Ted Borer is the Energy Plant Manager for Princeton University, and he advised us on the usability of the website. He mentioned that he uses electricity and firewood to heat his home, which caused us to add text to the website explaining this situation to users. Mr. Borer also recommended that we emphasize actions that are easily achievable.

Elke Weber 2 (4/18)
A key result from this follow-up meeting with Elke was that it would be nice to have pictures corresponding to "gas", "heating", and "electric" on the inputs page in order to make everything a bit less plain while maintaining objectivity. She also encouraged us to have two separate "About" pages: one for the policy (now the "Policy" page) and one for the implementation of the calculator (the "About" page).

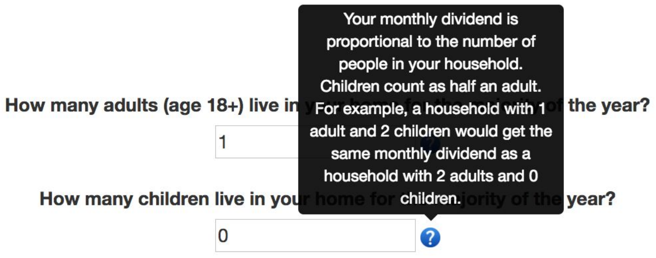
ACM 2 (4/20)
One of our users was the parent of an attendee, and his primary concern on the input page was that he has a daughter currently in college who is home for part of the year and he did not know whether he should include her in "children", in "adults", or in neither. We updated the site to specify that adult is 18+, and that "live in your home" means "for the majority of the year". Another very frustrating observation was that, despite that all four sets of users we interviewed said they were confused, none of them hovered over / clicked on the blue question marks to get further information. This inspired us to move more of the help text from the question marks to directly onto the page. Finally, we were inclined to add more descriptive text to prepare the user for where they will be going (e.g. "Wondering what you can do to decrease your tax? Click below to see how changing your energy, gasoline and heating usage habits could help lower your tax".

Design decisions

Principles

1. Transparency

The first central principle to our webpage is transparency, which means that information about what goes on under the hood with our app is not only available, but easily accessible and digestible. This means we will be keeping our code open source on GitHub. It also means we are careful to provide reiterative help text explaining what the app is doing under the hood, with plenty of concrete examples so that users can easily understand what we mean. We include information both in our “Policy” page and in the blue question marks next to the input boxes. We went through many iterations with users to get the language just right.



Your monthly dividend is proportional to the number of people in your household. Children count as half an adult. For example, a household with 1 adult and 2 children would get the same monthly dividend as a household with 2 adults and 0 children.

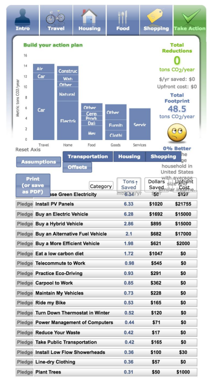
How many adults (age 18+) live in … the year?

1

How many children live in your home for … the year?

0

2. Nonpartisanship
   Our second guiding principle is nonpartisanship. This meant that we diverged from carbon footprint calculators and most other carbon tax calculators on quite a few design choices. First, we used a blue and black rather than green color scheme, and didn’t include any cliché pictures of leaves, trees, or billowing smoke clouds. We avoided any conversation about the environmental reasons for the policy, such as displacement of people from coastlines, damage from stronger storms, and health effects from pollution. Instead, we focused exclusively on the impact to the individual user’s wallet.
3. Simplicity
   Our third guiding principle is simplicity, which we hope increases likelihood of correct usage of our app. We were originally inspired by the Citizens Climate Lobby on this point, who have a simpler website, as compared to an overwhelming site from the CoolClimate Initiative (see right).



Total Reductions 0 tons CO2/year

$/yr saved: $0

Upfront cost: $0

Total Footprint 48.5 tons CO2/year

0% Better … household in United States with average size and similar income.

	Category	Tons Saved mtCO2e/yr	Dollars Saved $/yr	Upfront Cost $
	ase Green Electricity	6.34	$0	$127
Pledge	Install PV Panels	6.33	$1020	$21755
Pledge	Buy an Electric Vehicle	6.28	$1692	$15000
Pledge	Buy a Hybrid Vehicle	2.86	$895	$15000
Pledge	Buy an Alternative Fuel Vehicle	2.1	$682	$17000
Pledge	Buy a More Efficient Vehicle	1.98	$621	$2000
Pledge	Eat a low carbon diet	1.72	$1047	$0
Pledge	Telecommute to Work	0.98	$545	$0
Pledge	Practice Eco-Driving	0.93	$291	$0
Pledge	Carpool to Work	0.85	$362	$0
Pledge	Maintain My Vehicles	0.73	$228	$0
Pledge	Ride my Bike	0.53	$165	$0
Pledge	Turn Down Thermostat in Winter	0.52	$120	$0
Pledge	Power Management of Computers	0.44	$71	$0
Pledge	Reduce Your Waste	0.42	$17	$0
Pledge	Take Public Transportation	0.42	$165	$0
Pledge	Install Low Flow Showerheads	0.36	$100	$30
Pledge	Line-dry Clothing	0.36	$57	$0
Pledge	Plant Trees	0.31	$50	$1000

   Removing a lot of lengthy explanatory text (we went from a full page to a 4-sentence description on the Index view) in favor of concrete examples for the sake of transparency helped us achieve this simplicity. From a structural standpoint, we kept all inputs on a single page, so that ultimately the user only has to navigate from the welcome page, to an input page, to a results page, to an actions page. We also decided to have only three of the most important actions on the Actions page - one for each of electricity, gasoline, and heating.

   From a visual standpoint, we also bolded the most important text, along with bolding the color of the key information on each graph (in particular, net cost and net benefit respectively for the graphics on the results and actions pages). The red or green colors of the graphics also correspond to negativity or positivity, and change interactively on the actions page. We have a visually striking “Next” button on each page for obvious navigability.

Finally, on the input page, an important feature is that average values are already filled out, so that if a user isn’t sure about their energy use they can guess based on how much more or less they use relative to average. Although this reduces accuracy, based on user feedback the ameliorated ease of use is well worth the cost.

4. Privacy
   Our final guiding principle is privacy. Informed by the recent Cambridge Analytica scandal, we decided not to store any user information on our own servers. Although we began by saving user data into our database, we ultimately switched to using cookies for saving a users info on their own browser.

Languages and Systems

1. Backend: Django
   Jonathan already had experience with Django through Princeton Math Competition. We also all love Python. So a fundamental axiom was that we would use Django for the framework. In addition to the obvious benefits (like automatically getting our models talking to our views and our URLs redirecting with ease), it also made implementing cookies a breeze: setting and getting cookies is a one-line endeavor.

2. Frontend: Bootstrap, JavaScript
   We use Bootstrap to generate our front end design, leveraging its robustness to browser settings and automatic resizing. We chose to use JavaScript libraries jQuery and Chart.js to generate the sliders and graphs (respectively) on our website. Finally, for implementing less cookie-cutter features like “stickiness” of our graphics to the bottom of the header label on the Actions page, we made JavaScript functions from scratch.

3. Deployment: Heroku
   Since it was free, well-documented, and had a comprehensive tutorial, we chose to use Heroku to deploy our site. This in turn guided us to choose PostgreSQL for our database.

4. Database: PostgreSQL
   Heroku works particularly well with PostgreSQL, and the tutorials we used for Heroku all used this database. So this was an obvious choice for us.

5. Version Control: GitHub
   This was required, and was very useful for version control.

Advice to future 333 groups

1. Come in with a full project idea in mind and a minimum viable product.
2. User research is key. Do it early and continue throughout the semester, before you build something no one will use.
3. Set concrete deadlines and achievables for every meeting, and every week.
4. Work on important problems: distinctive (read: off-campus), interesting, and meaningful.
5. A well-done small product is better than a poorly-done larger product. Make sure to do the basic things well, such as descriptive text and eliminating dumb bugs. User feedback was hampered by noticing the painfully obvious things which people can't look past to the core, "interesting" features.
6. Don't be dismayed by the steep learning curve in the beginning: bash through it by continuing to meet consistently and often.
7. Keep a log with To-Dos at the top, and a summary of accomplishments from every meeting.