SUSTAINABLE JERSEY® • CERTIFIED •
A Better Tomorrow, One Community at a Time

2017 NEW JERSEY
SUSTAINABILITY
SUMMIT

FOR SCHOOLS
SUSTAINABLE JERSEY® • CERTIFIED •
A Brighter Future, One School at a Time

Welcome to
Coding for Community Demonstrations

WiFi network: Guest-at-TCNJ3 | Username: guest1783 | Password: ujy6uvuh

SUSTAINABLE JERSEY
CERTIFIED
A Better Tomorrow, One Community at a Time

CODING FOR COMMUNITY

A CIVIC TECH COMPETITION TO SOLVE LOCAL ISSUES

2017

“

All change is **hard** at first,
messy in the middle,
and **gorgeous** at the end.

-Robin Sharma

CFC 2017

50+ municipal needs

30+ project proposals

11 finalists (1,2,3Hon.+8)

THE RESULTS ARE IN

- 1st Place - Encore Dev Labs
  raspberry pi temperature tracker
- 2nd Place - Dashability
  SJ actions real-time dashboard
- Hon. Mention (innovation)
  Troop 58 (My Town Hall)
- Hon. Mention (creative data use)
  Bike View, by Bike View
- Hon. Mention (presentation)
  Hansen Unlimited (street lamps)
- 8 other 2017 winners
  (find at cfc.sustainablejersey.com)

and a bright future?

and a bright future?

*who wants to **help build** it?*

“

the new leadership is about working together.

-Jack Ma

SUSTAINABLE JERSEY
CERTIFIED
A Better Tomorrow, One Community at a Time

CODING FOR COMMUNITY

A CIVIC TECH COMPETITION TO SOLVE LOCAL ISSUES

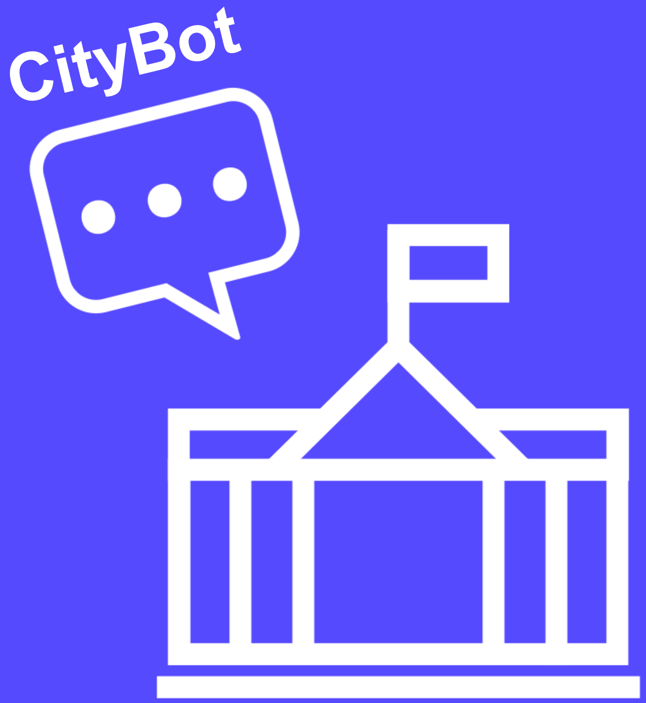
2017

My Town Hall

- Troop 58
  - John Fleming
  - Cameron Fleming
  - Rowan Johnson
  - Chris Algieri (*not present*)
- slides

Street Lamp Detection

- Hansen Limited
  - Aruj Jain
  - Gustav Hansen
- slides


CityBot

New Brunswick Office of Innovation Challenge

Develop a chat bot to municipalities to:

1. Public access to municipal services
2. Public engagement with city government
3. Municipal awareness of public opinion
4. Access to voters and their discontent

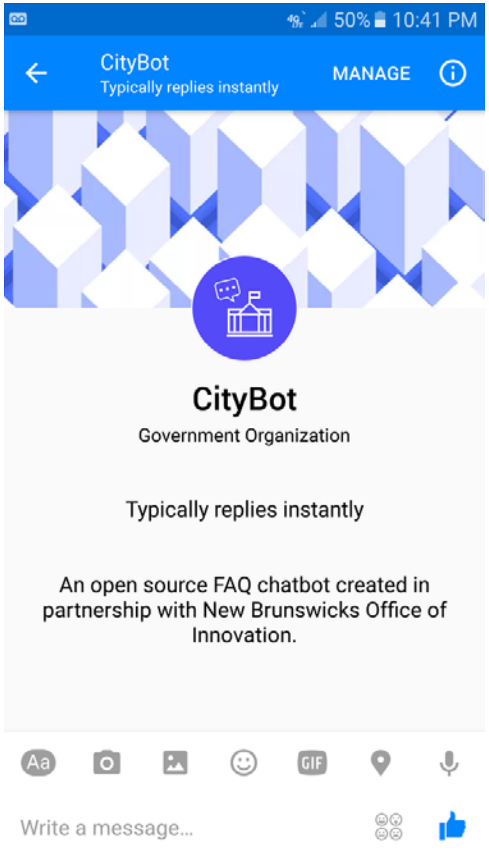
Engage constituents where they are:

1. 60% of Americans are on Facebook Messenger in 2015*
2. Messaging apps occupy 91% of all time spent on mobile phones in 2015**

* http://www.theverge.com/2015/9/4/9264069/facebook-messenger-google-youtube-apps
** https://qz.com/596981/whatsapp-is-finally-making-a-key-move-that-will-change-how-we-use-chat/

Facebook Messenger bots can give answers to someone asking…

1. … when public meetings are
2. … about city services
3. … what facilities they can contact
4. … whatever else you choose to add!

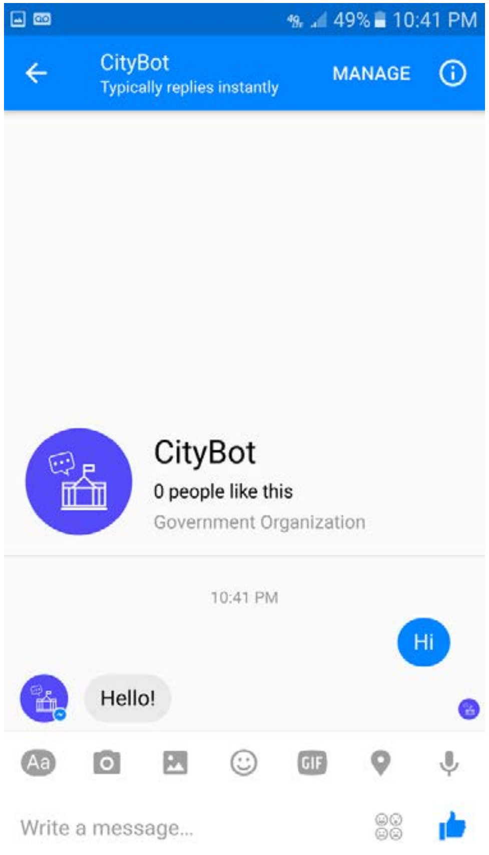


CityBot

Government Organization

Typically replies instantly

An open source FAQ chatbot created in partnership with New Brunswicks Office of Innovation.



CityBot

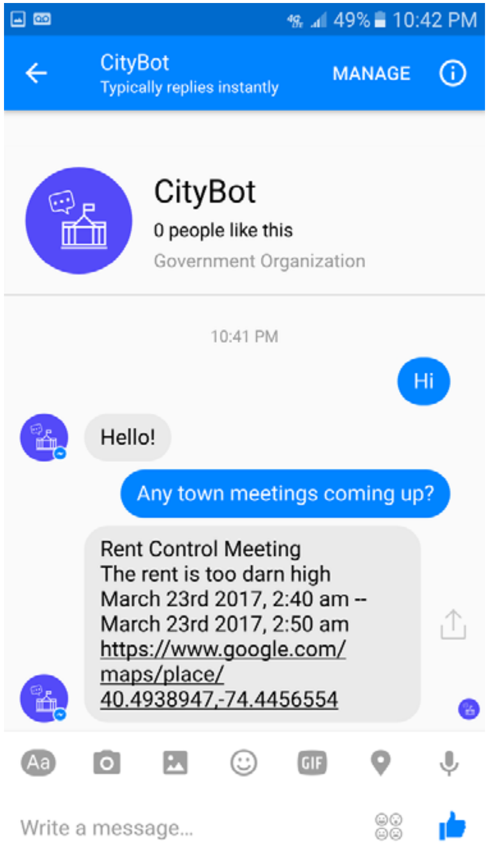
0 people like this

Government Organization

10:41 PM

Hi

Hello!



CityBot

0 people like this

Government Organization

10:41 PM

Hi

Hello!

Any town meetings coming up?

Rent Control Meeting
The rent is too darn high
March 23rd 2017, 2:40 am --
March 23rd 2017, 2:50 am
https://www.google.com/maps/place/40.4938947,-74.4456554

Short-term Impact

Handle repetitive requests with instant answers!

Medium-term Impact

Understand constituents engaging the bot

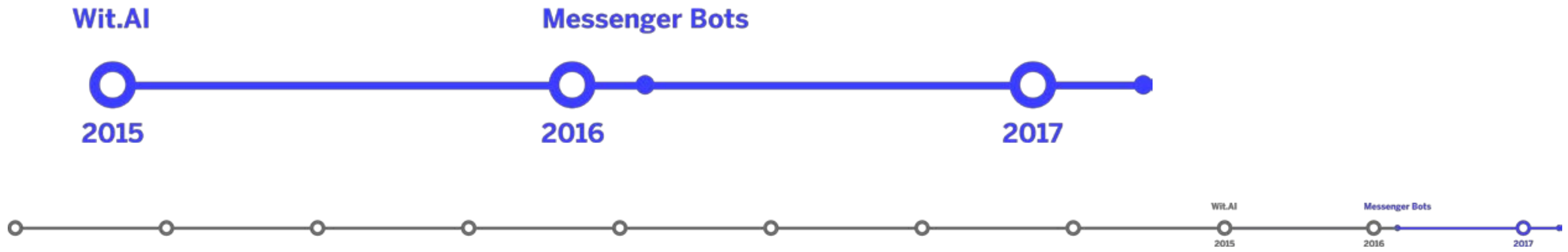
Long-term Impact

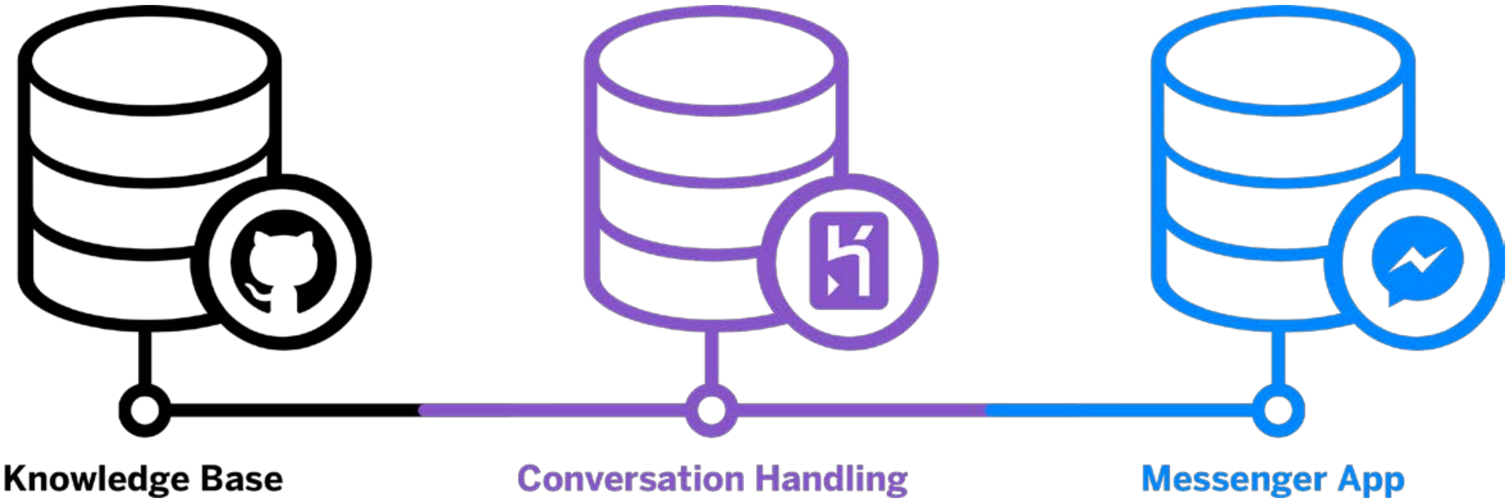
Lead a new era of constituent engagement!

Does any local gov have a bot?

Singapore (contact directory), San Paulo (service requesting), and Los Angeles (business portal). All custom built.

City Bot gives a framework for you to launch your own.




Knowledge Base
Conversation Handling
Messenger App

README.md

CityBot

CityBot is a Facebook Messenger bot that is easy to spin up and plug into your local government Facebook Pages. If you're not a developer, don't worry :) There's no coding involved, just a bit of configuration legwork at the start.

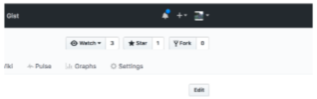
Setting up Your Bot

1. Copy the Project
2. Launch the bot
3. Connect the bot to a Facebook
4. Write your first answers
5. Expand what the bot can do!

Copy the Project

So you can configure your application however you need, you're going to need your own instance of the CityBot. That starts with simply copying it, or in Github lingo, "forking" it.

To do that, simply click the button in the top right, and that's it!

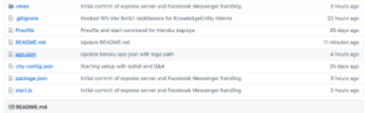


Launch the Bot

Now that you're in your *forked project*, let's get this code online. Here's what we need to do:

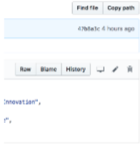
1. Edit `app.json`
2. Deploy to bot to Heroku
3. Tell Heroku to restart the bot when updates are made

Editing app.json



views Initial commit of express server and Facebook Messenger handling 3 hours ago
.gitignore Hooked Wit into Botkit middleware for KnowledgeEntity intents 22 hours ago
Procfile Procfile and start command for Heroku deploys 25 days ago
README.md Update README.md 11 minutes ago
app.json Update heroku app json with logo path 4 hours ago
city-config.json Starting setup with botkit and Q&A 25 days ago
package.json Initial commit of express server and Facebook Messenger handling 3 hours ago
start.js Initial commit of express server and Facebook Messenger handling 3 hours ago
README.md

In the files listed at the top, simply click on `app.json`. When viewing the file, you will be able to edit it by clicking the little pencil on the right side of the toolbar. Once in edit mode, you must change the value of the "respository" field to reflect this URL. Look at the URL in your browser bar for this. In the end you'll be changing



```
{
  ...
  "repository": "https://github.com/Unit
  ...
}
```

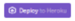
to something, for example, like

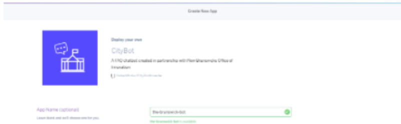
```
{
  ...
  "repository": "https://github.com/my_u
  ...
}
```

Feel free to change the name and description as well, but that is not required.

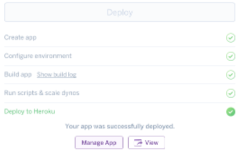
Deploying to Heroku

Once you've done that, just click this button! If you do not have an account with Heroku, you will be asked to make one. Don't worry, there is a free plan!



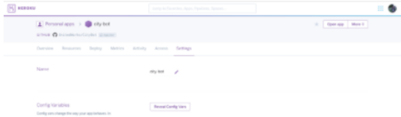


Before you see a successfully running application, we just need to do one thing! Click the `Manage App` once Heroku has successfully set up the project.



Heroku Magic

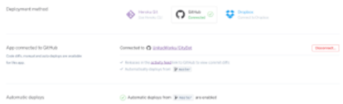
For our application to work, we need to tell our bot how to speak. Luckily United Works and the New Brunswick Office of Innovation are doing that for you. In the settings of your newly setup application, you'll see an option to set "Config Variables". There we want to add some KEYs and VALUEs.



The first KEY will be `WIT_TOKEN`, but you will need to contact me (mark@mayor.chat) for the value because of security reasons. The second KEY is `PORT`, which can have the VALUE `8000` for now. The third KEY will be `FACEBOOK_VERIFY_TOKEN`, and the value can be whatever you want, even something like `hamsandwich`. We're going to tell Facebook this phrase later so their servers can talk to our bot, so remember it!

Once those are set, the server will automatically update and restart.

Lastly, since we're here, it will be a huge convience for us to link the bot to our copied project in Github so the bot automatically updates when changes are made. To do that, go to the "Deploy" page and select "Github" from the "Deployment Methods". Find the correct account and repo from their menu and make sure `Automatic deploys from master are enabled`.



Great! Now simply click "Open App" in the upper right to see a screen telling you the server is started!

FYI for Programmers

At this point, you will be able to run the application locally using `npm start`, and chat with the bot via the terminal.

Connect to Facebook

1. Get Access to the Tools
2. Tell Facebook about your App
3. Tell Heroku about Facebook

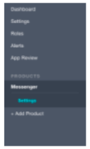
Get Access to the Tools

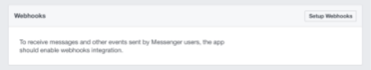
All this means is logging into Facebook's developer portal! https://developers.facebook.com

Telling Facebook about your App

Once logged in, you're going to want to create a new application. It's only a few questions that you can answer however you like. Once you've done that, there are a few things we need to do.

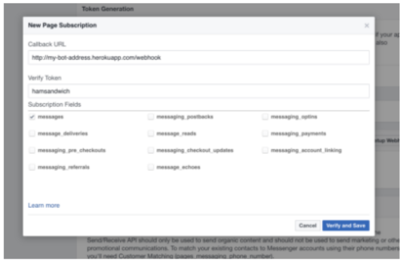
First, in the lefthand sidebar, go to "Messenger" tab under "Products". We're going to tell Facebook where our bot's server is. To do that, scroll down to "Webhooks" and click "Setup Webhooks" which will bring up a screen with two inputs and some checkboxes.





Webhooks
Setup Webhooks

To receive messages and other events sent by Messenger users, the app should enable webhooks integration.



Token Generation

New Page Subscription

Callback URL: http://my-bot-address.herokuapp.com/webhook

Verify Token: hamsandwich

Subscription Fields

☑ messages ☐ messaging_postbacks ☐ messaging_optins
☐ message_deliveries ☐ message_reads ☐ messaging_payments
☐ messaging_pre_checkouts ☐ messaging_checkout_updates ☐ messaging_account_linking
☐ messaging_referrals ☐ message_echoes

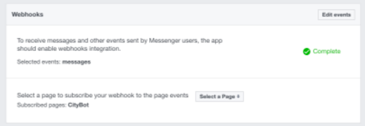
Learn more

Cancel Verify and Save

Send/Receive API should only be used to send organic content and should not be used to send marketing or othe promotional communications. To match your existing contacts to Messenger accounts using their phone numbers you'll need Customer Matching (pages_messaging_phone_number).

For callback URL, you're going to want to put whatever your server's address was, and append `/webhook`. That should end up looking something like `https://new-brunswick-bot.herokuapp.com/webhook`. For the Veriy Token, you're going to type in that secret phrase before, where mine was `hamsandwich`. If you can't remember it, look at the "Config Variables" on your bot's server settings. Lastly check off the box for "messages". Now just click "Verify and Save"

Once that goes through, select what page you want Facebook to listen to for messages and forward to your bot.



Webhooks
Edit events

To receive messages and other events sent by Messenger users, the app should enable webhooks integration.

Selected events: **messages**

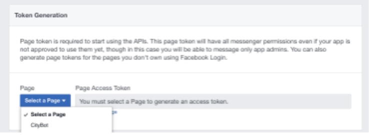
Complete

Select a page to subscribe your webhook to the page events Select a Page

Subscribed pages: **CityBot**

Telling Heroku about Facebook

So our bot knows how to respond to Facebook, we need a little piece of information. Under the Messenger panel where we set up our webhook, we need to generate a token for our page. All that means is selecting from the drop down which page you want the bot to be. Copy the long set of characters and numbers that appear next to that drop down.



Token Generation

Page token is required to start using the APIs. This page token will have all messenger permissions even if your app is not approved to use them yet, though in this case you will be able to message only app admins. You can also generate page tokens for the pages you don't own using Facebook Login.

Page: Select a Page (✓ Select a Page, CityBot)

Page Access Token: You must select a Page to generate an access token.

Now, back on your Heroku server's "Config Variables", we're going to add a KEY/VALUE for that long string of characters and numbers. The KEY will be `FACEBOOK_PAGE_TOKEN` and the VALUE, whatever Facebook gave you for that page.

Take a deep breath

Ready to give it a spin? Click "App Review" from the sidebar of the Facebook Developer panel, and switch your app to being "Public" at the top! BUT **DON'T WORRY! Only** you will receive responses

DON'T WORRY! Only
from the bot. The pub
to-person chat interfa
your bot, you will have
page and be approved

Go to the Facebook p
for, and send a messa
:)

Hello!

Recycling is every other Tuesday

Write your First Answ

You did it!!! I'd give yo
you'll just have to take

Go through any availa
`base` directory, and e
your town or city. Unle
programming, always
quotations are right! A
for "intent". Our langu
phrase and expects a

```
...
{
  intent: 'GAR
  answer: 'Tra
},
...
```

If you want to experim
with language process
`catchPhrases` key.

```
...
{
  catchPhrases
  answer: 'Hel
},
...
```

When you edit, make
"," at the end of each
opening brace "{"

Expand the Bot

As time goes on, we w
knowledge base files
to be handled. We sug
repository for those u

If you have any sugge
you want this bot to h
know over at this issu
https://github.com/Ur

Enjoy :)

CityBot’s Problem

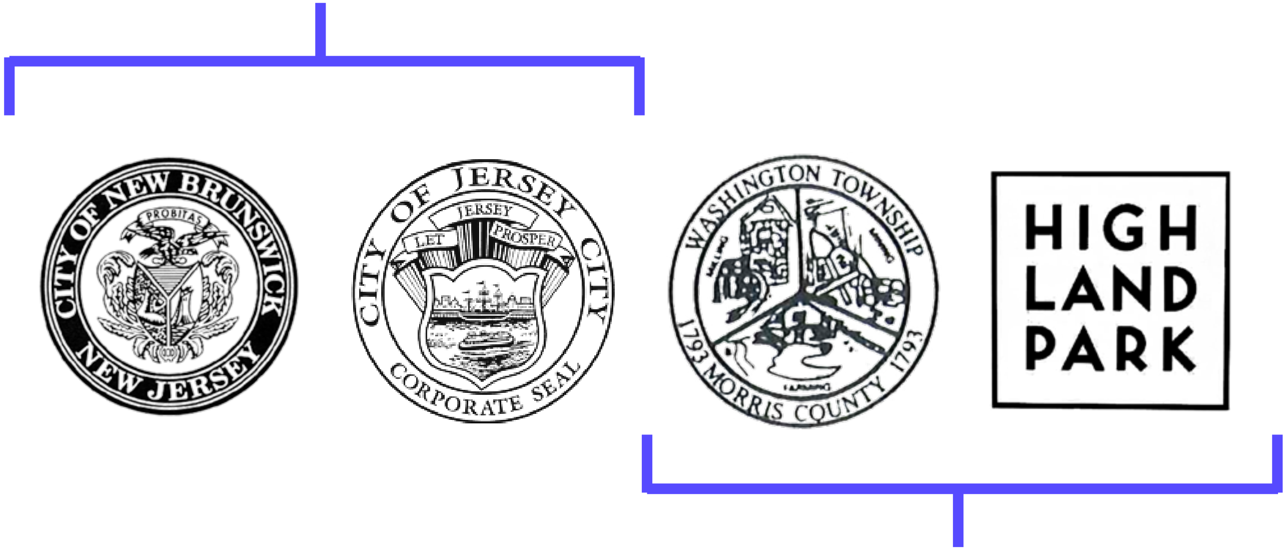
Do you have a Github account?

Have you touched code?

Brendan has developer experience
NBOI has access to resources

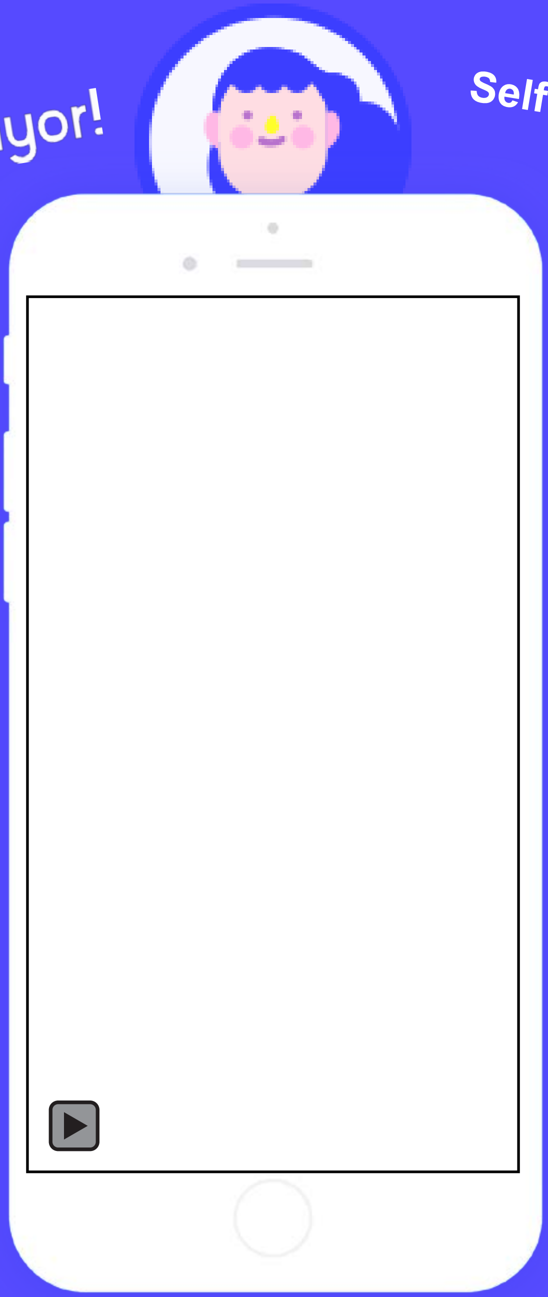
CITY OF NEW BRUNSWICK
PROBITAS
NEW JERSEY

Access to resources
Some access to developers



Little or no resources
No developers on staff

Hey Mayor!



Self-service for towns

What We’ve Learned

1. Messenger is understood for communication, not navigating processes/information (yet).
2. Perfect initial conversational flow to let constituents realize what they can ask.
3. Answers aren’t everything constituents want. Some folks are bored and want entertainment. Some folks are annoyed and need to have expectations communicated.

Current Challenges: Content, Content, Content

1. Who wants to answer 300 questions?
2. “Just call us” or “Call them” answers
3. Integrating into workflows

A small ask…

Tell us your stories and challenges with constituent communication!

mark@mayor.chat

Bike view

data, visualizations, safety

Hema Waghray – Code For Princeton
Tineke Thio – Princeton Bicycle Advisory Committee

Code For Princeton – visualizations of crash data

Code for Princeton is about #civictech. We work to make government accessible and easy for citizens.

We create technology solutions to real world problems. We are like scouts on steroids.

Princeton – Bicycle Master Plan

- Connected Network of Bicycle Routes
- Low level of traffic stress

Bike View User Interface

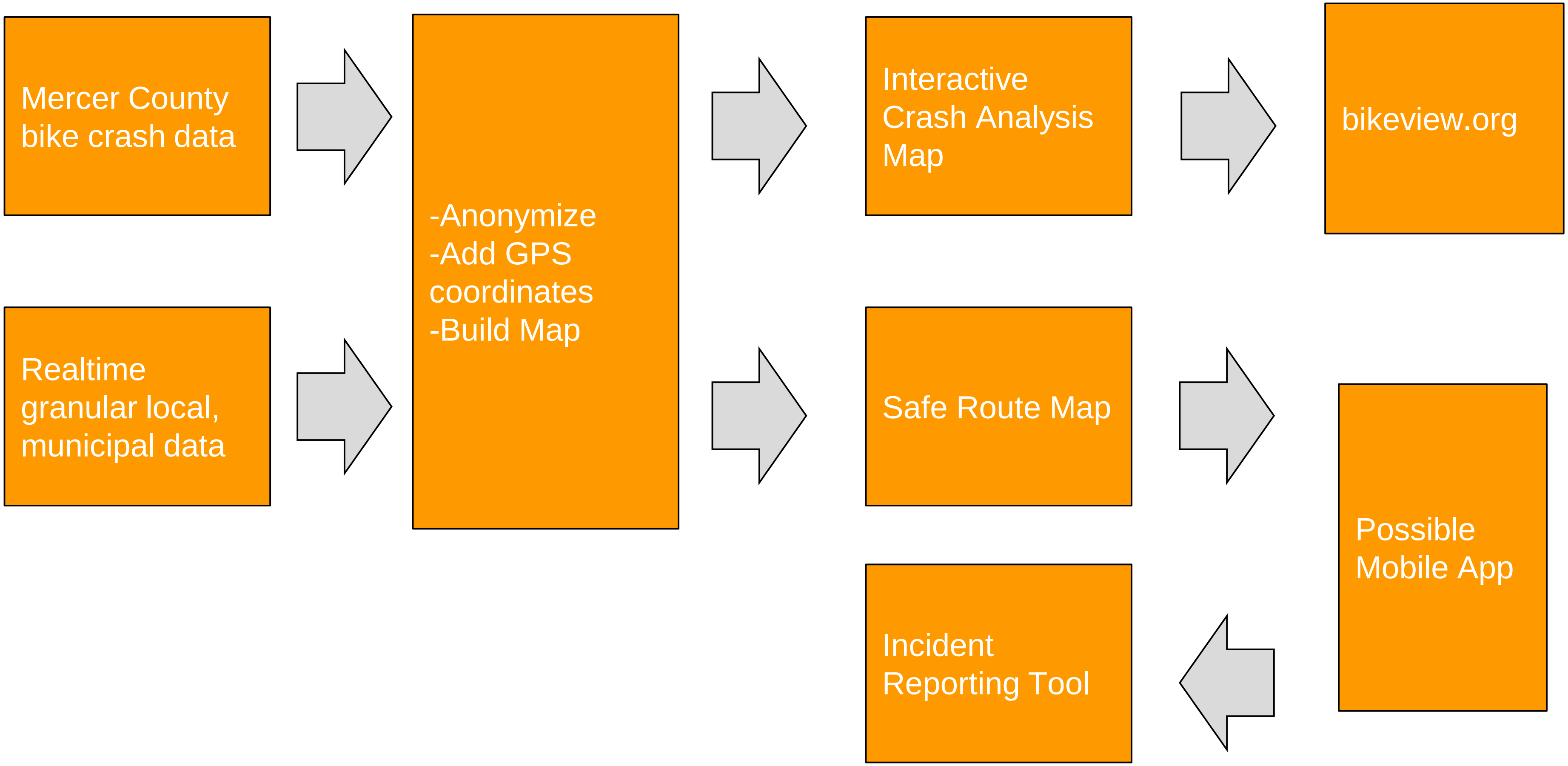
Original code for data analysis

Crash Statistics turned into interactive maps and visualizations - municipality and planners

Easy access to find safe routes based on crash data

It is meant for the public - open source data

Data ► Processing ► Application ► Presentation



Mercer County bike crash data

Realtime granular local, municipal data

-Anonymize
-Add GPS coordinates
-Build Map

Interactive Crash Analysis Map

Safe Route Map

Incident Reporting Tool

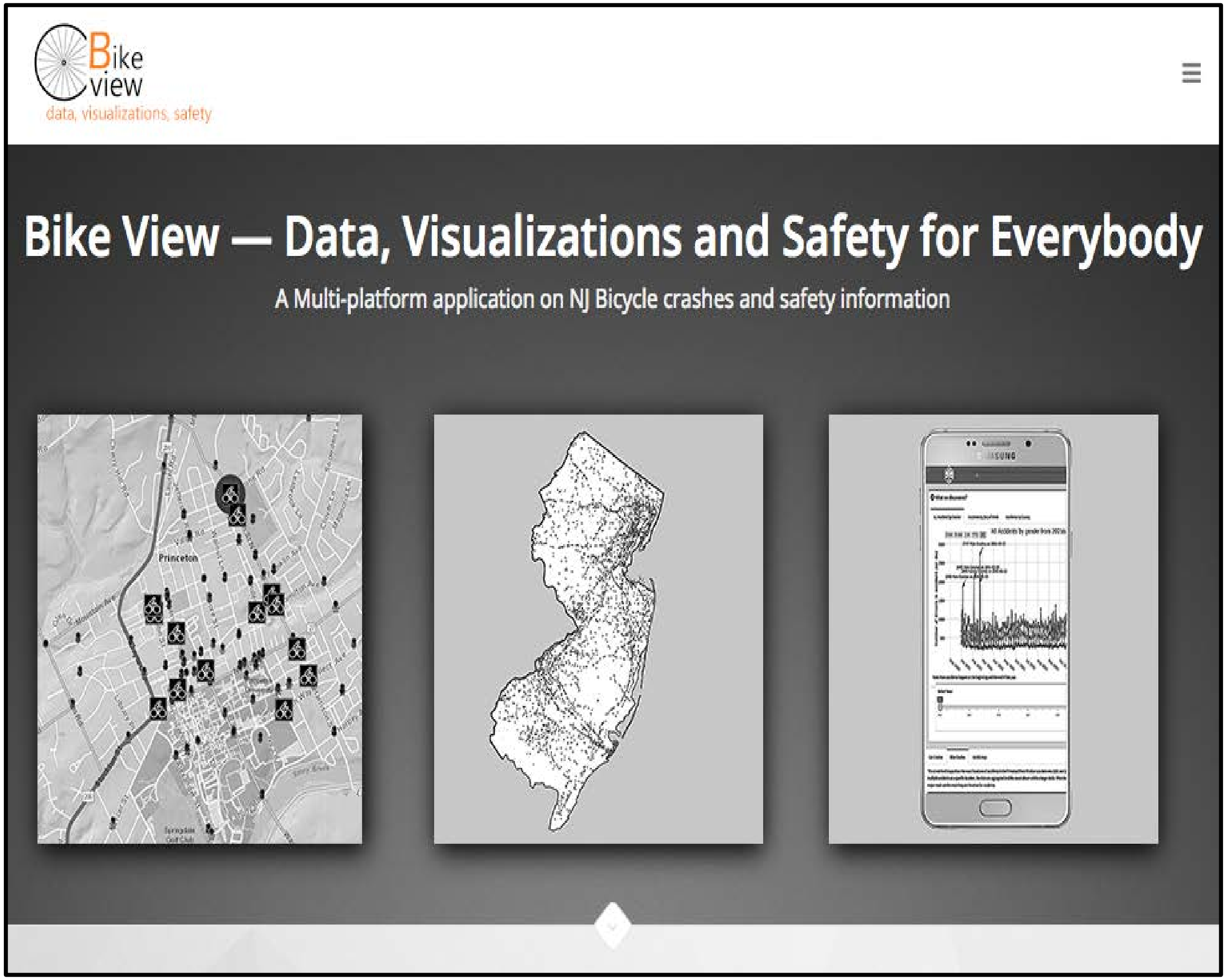
bikeview.org

Possible Mobile App

Technical Toolbox

- Data Analysis and Visualization - R, Shiny App, Plotly & D3
- Website - HTML5, CSS, PHP and MySQL
- Mobile app - Android SDK
- Mapping - CartoDB, ArcGIS, Google Geocoding API

Team’s solution: bikeview.org



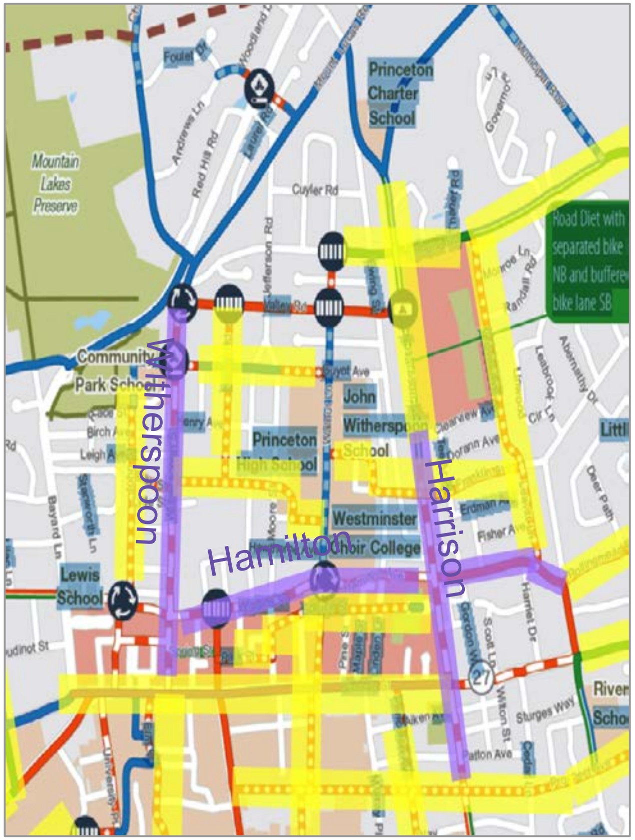
Bike View — Data, Visualizations and Safety for Everybody

A Multi-platform application on NJ Bicycle crashes and safety information

Car crash map

Early Success: Stronger Bicycle Master Plan

Princeton Bicycle Master Plan
Low-traffic-stress routes highlighted



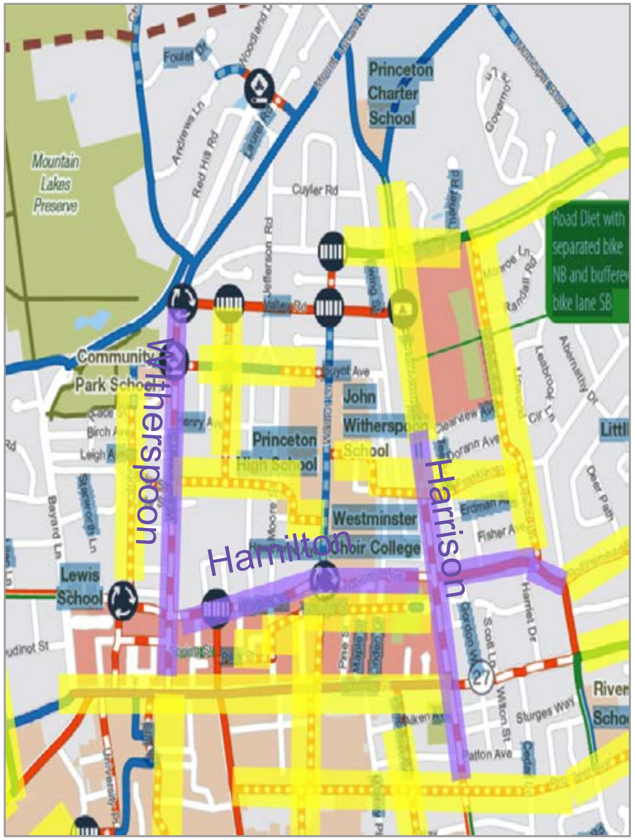
Gaps in safe network

Early Success: Stronger Bicycle Master Plan

Bicycle Master Plan:

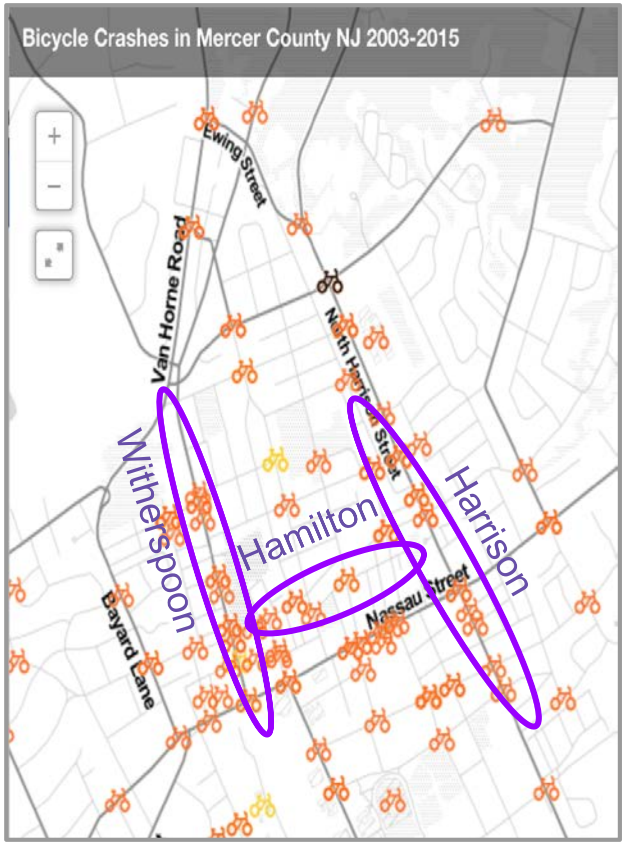
PBAC recommends higher priority on problem corridors

Princeton Bicycle Master Plan
Low-traffic-stress routes highlighted



Gaps in safe network

Bike View bike crash map

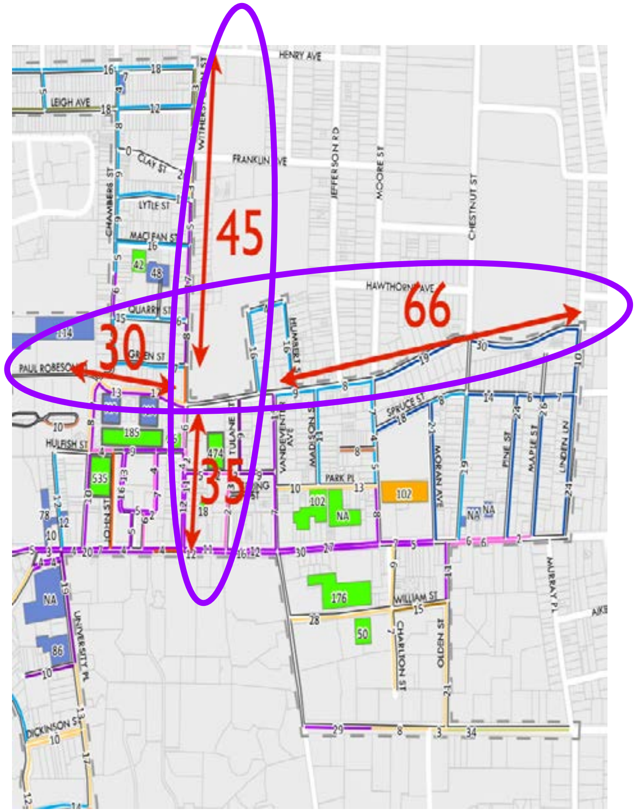


High crash corridors

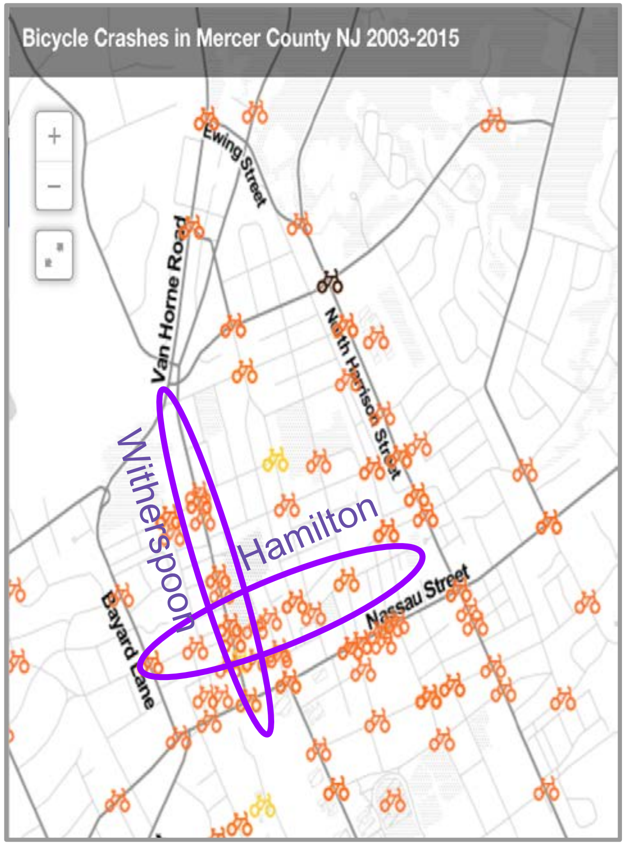
Current Impact: Princeton Parking Study

On problem corridors PBAC recommends:

- Remove <span style="color:red">176 on-street parking</span> (2.5% of 7,000 total)
- Replace with bicycle lanes

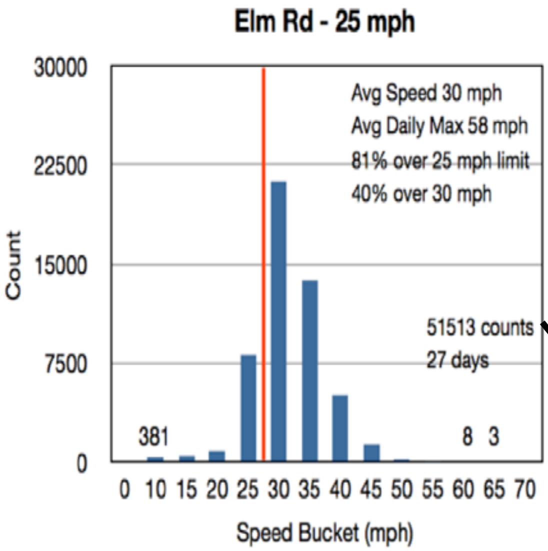


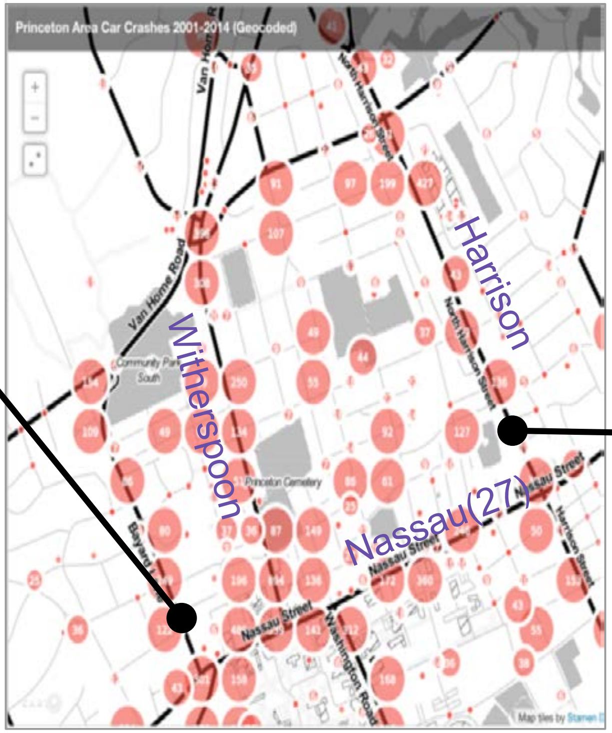
Downtown Princeton Parking Inventory
Total about 7,000 spaces



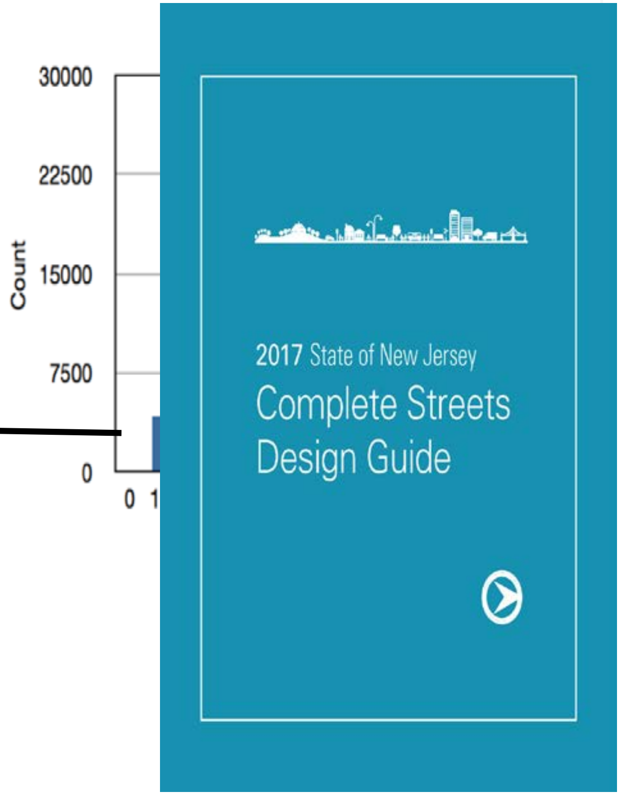
Bike View bike crash map

This Fall: Traffic Calming & Complete Streets





Interactive car crash map



2017 State of New Jersey
Complete Streets
Design Guide

“Scouts on Steroids Help NJ Town”

Volunteer technologists
“Mission Impossible” team – diverse talents
Cross-disciplinary collaboration

Complete Streets
Access For All

@codeforprinceton
@PrincetonBike

Hyperlinks

bikeview.org

Shiny App Dashboard

Bikes-only Crash Map

Cars-only Crash Map

Mobile App

Map demo/tutorial video