

Human-Computer Interaction Design



COGS120/CSE170 - "Intro. HCI"

Instructor: Philip Guo

Lab 7 - Analytics and A/B Testing (2016-11-10)

by Michael Bernstein, Scott Klemmer, and Philip Guo

Analytics can answer how and which questions

- “How are people are using my design?”
- “Which design should I pick?”

Core actions in Google Analytics

- Record *pageviews*: how many people are viewing this page?
- Record *events*: who clicked on this button, or uploaded a picture?
- Run *experiments*: which alternative leads to more pageviews or certain events?

Using Google Analytics

1. Register your site with Google Analytics
2. Insert Google's JavaScript in your HTML/handlebars files
3. Look at Google Analytics website to see the results.

NOTE THAT THERE IS A 24-HOUR TIME DELAY AND THERE MAY BE BUGS WHEN YOU SET IT UP THE FIRST TIME, SO DON'T DO YOUR ANALYTICS ASSIGNMENT LAST-MINUTE!!! You won't be able to get TA help last-minute on this part since Google delays data by ~24 hours.

Including Google Analytics JavaScript code

(this is just demo code ... real code is later)

Include the Javascript block on every page in your site, so it records who visited which page

```
(function(i,s,o,g,r,a,m){i['GoogleAnalyticsObject']=r;i[r]=i[r]||function(){(i[r].q=i[r].q||[]).push(arguments)},i[r].l=1*new Date();a=s.createElement(o),a.async=1;a.src=g;m.parentNode.insertBefore(a,m)})(window,document,'script','//www.google-analytics.com/analytics
```

```
ga('create', 'UA-XXXXX-Y', 'auto');  
ga('send', 'pageview');
```

Including Google Analytics JavaScript code

(this is just demo code ... real code is later)

“Minified”
logging code
that Google
gives you

```
(function(i,s,o,g,r,a,m){i['GoogleAnalyticsObject']=  
(i[r].q=i[r].q||[]).push(arguments)},i[r].l=1*new  
m=s.getElementsByTagName(o)[0];a.async=1;a.src=g;m  
})(window,document,'script','//www.google-analytics
```

Register account;
‘auto’ means works
on both localhost
and Heroku

```
ga('create', 'UA-XXXXX-Y', 'auto');
```

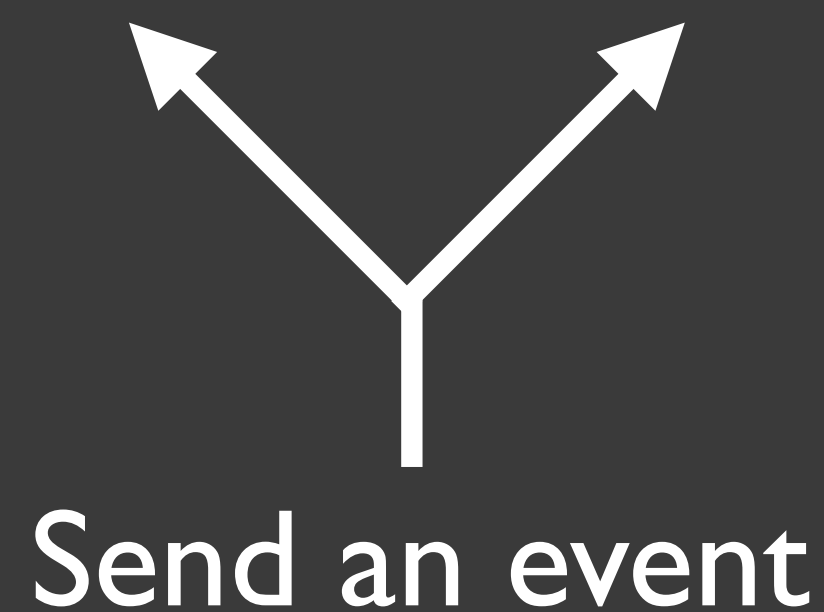
Send
page view
to server

```
ga('send', 'pageview');
```

Record an event

Any outcome that you care to record: leaving a comment, clicking on an image, scrolling to the bottom of the page...

```
ga('send', 'event', 'button', 'click');
```



Category:
the object that the user
interacted with



Action:
the behavior the user took

Store any kind of event

```
ga('send', 'event', 'friendsuggestion', 'dismiss');  
ga('send', 'event', 'friendsuggestion', 'accept');  
ga('send', 'event', 'pagescroll', 'bottom');
```

Additional arguments available if you want categories or numeric counts.

<https://developers.google.com/analytics/devguides/collection/analyticsjs/events>

A/B testing

A/B testing in three steps

1. On the server, create two different URLs to represent your two conditions. They can be the same controller.
2. Log events for any outcome variables you care about.
3. Create the experiment in Google Analytics and insert their experiment code into your HTML/handlebars.

Creating URL endpoints in app.js

```
// register the control (original) condition  
app.get('/', index.view);
```

```
// register the experimental condition  
app.get('/new', index.view2);
```

Send different data to the template

The controllers in index.js render the same template but send different data:

```
data = <some data object to pass into template>;
```

```
exports.view = function(req, res) {  
  data['showAlternate'] = false;  
  res.render('index', data);  
};
```

```
exports.view2 = function(req, res) {  
  data['showAlternate'] = true;  
  res.render('index', data);  
}
```

Render the alternatives using if/else statements in Handlebars

```
<button>
  {{#if showAlternate}}
    You should sign up for Intro to HCI NOW!!!
  {{else}}
    Sign up for Intro to HCI.
  {{/if}}
</button>
```

Log the event

In the client Javascript:

```
ga('send', 'event', 'signup', 'click');
```

One important limitation about Google Analytics: by default, it can track only binary events. That is, what % of people who come to your site clicked the sign-up button? If you want to track view counts (e.g., how many pictures did the person view?), it's very hard. You can do it using an API, but we don't recommend it for this class.

practice

Analytics

Goal: install Google pageview analytics
on your portfolio

Get the starter code

- Fork the repository: <https://github.com/pgbovine/lab7>
- `git clone` your forked repository into the `introHCI` directory as usual

Start node.js and visit <http://localhost:3000>

You should see the normal page with some extra Like buttons on the projects

Michael Bernstein

human-computer interaction · social computing · crowdsourcing

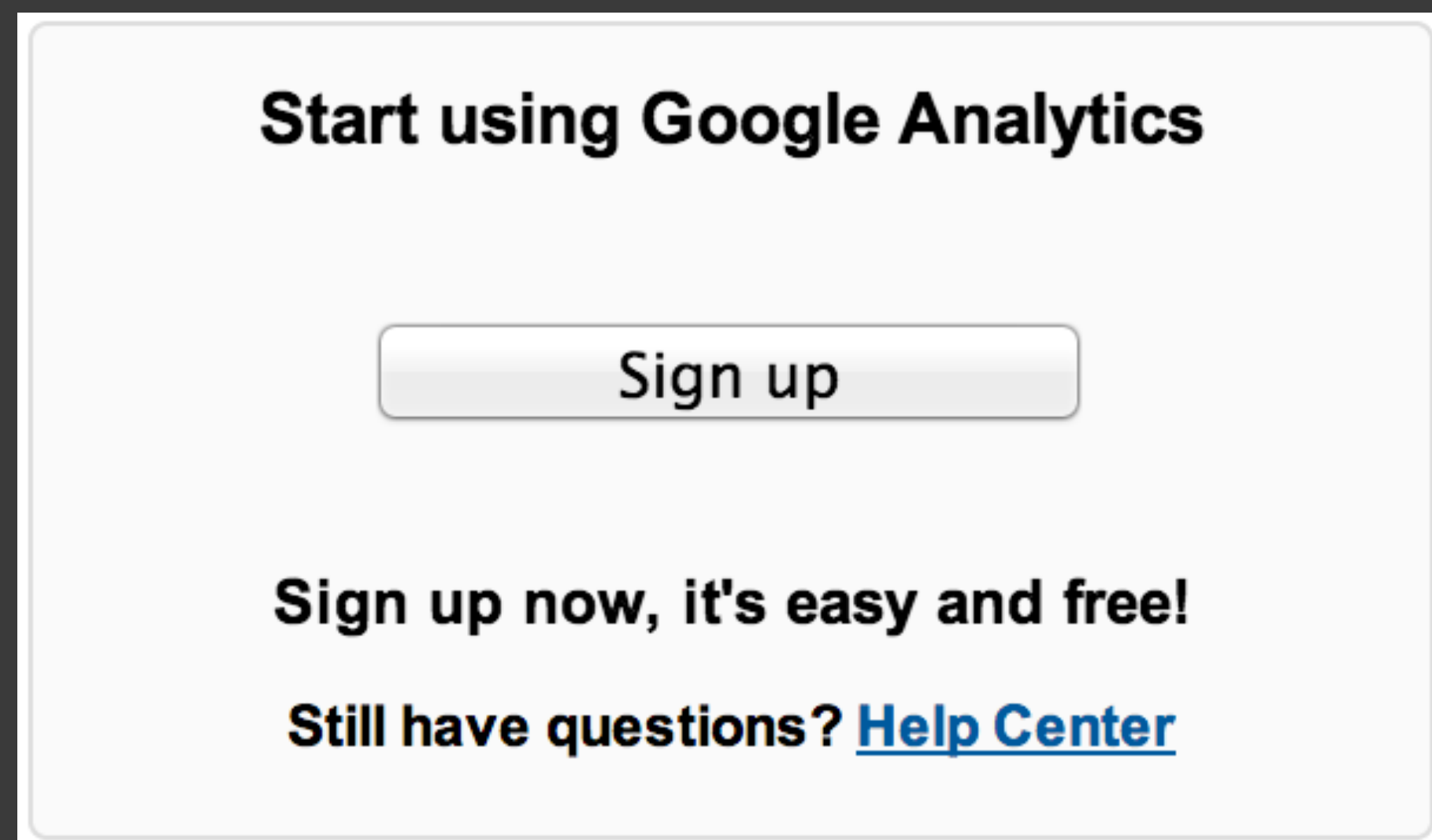
Projects



Open a Google Analytics account

<https://analytics.google.com/analytics/>

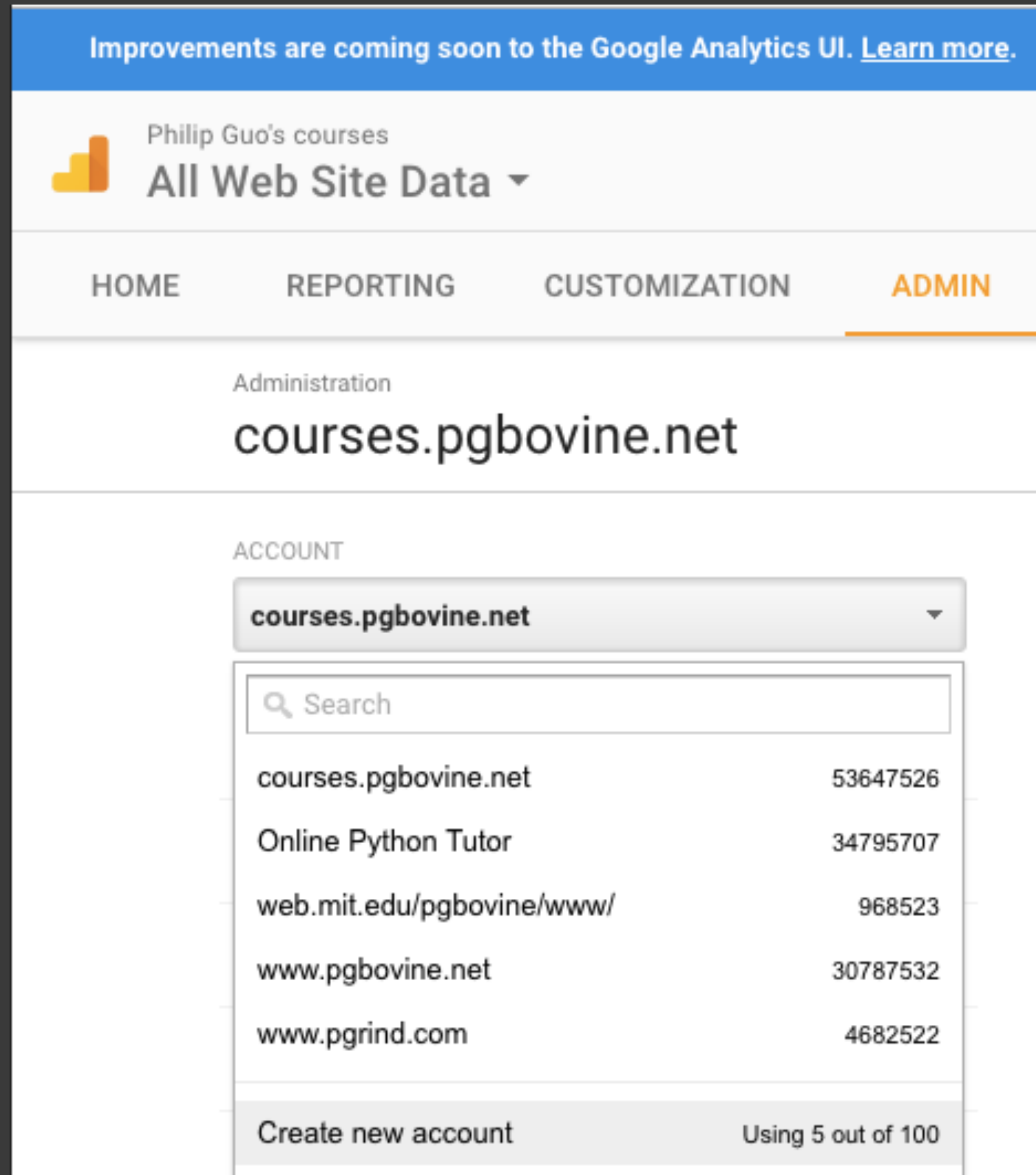
- Click “Sign in” (upper right)
- Create a Google account if you don’t have one:



- If you do have an account, see the next slide

If you already have a Google account

- You'll need to create a new analytics account and property:
 - Click the “Admin” tab at the top of the page
 - Click the dropdown under “Account” and choose “Create new account”



Improvements are coming soon to the Google Analytics UI. [Learn more.](#)

Philip Guo's courses
All Web Site Data ▾

HOME REPORTING CUSTOMIZATION **ADMIN**

Administration
courses.pgbovine.net

ACCOUNT

courses.pgbovine.net ▾

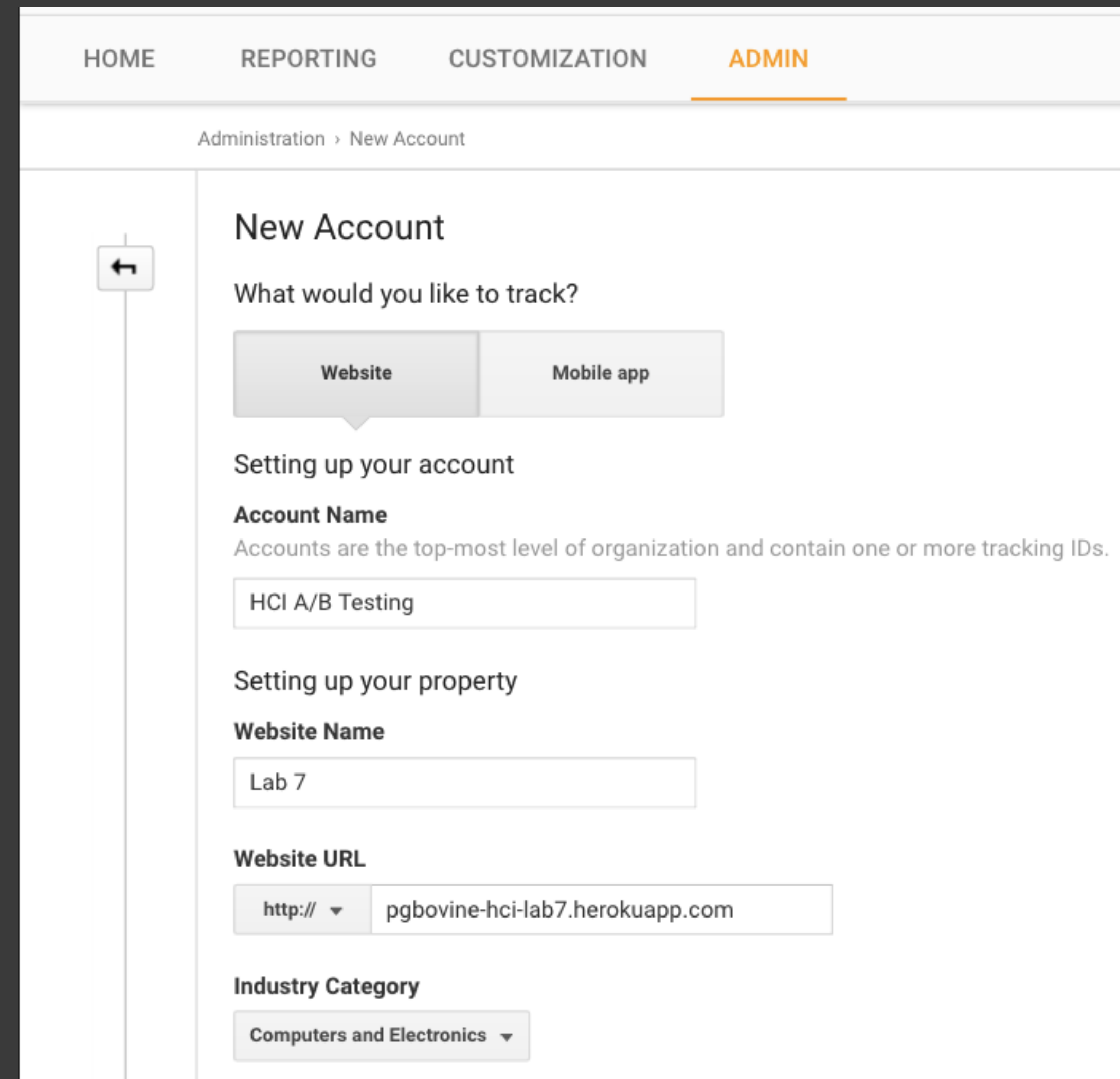
Search

courses.pgbovine.net	53647526
Online Python Tutor	34795707
web.mit.edu/pgbovine/www/	968523
www.pgbovine.net	30787532
www.pgrind.com	4682522

Create new account Using 5 out of 100

Set up the account

- Choose an account/website name can be “Lab 7” or similar
- Website URL will be your Heroku app’s URL, e.g., mine is
 - `http://pgbovine-hci-lab7.herokuapp.com`
- Click the “Get Tracking ID” button and accept the terms of service



The screenshot shows a web interface for creating a new account. At the top, there is a navigation bar with 'HOME', 'REPORTING', 'CUSTOMIZATION', and 'ADMIN' (highlighted). Below the navigation bar, the breadcrumb 'Administration > New Account' is visible. The main content area is titled 'New Account' and asks 'What would you like to track?' with two buttons: 'Website' (selected) and 'Mobile app'. Underneath, there are sections for 'Setting up your account' and 'Setting up your property'. The 'Setting up your account' section includes an 'Account Name' field with the value 'HCI A/B Testing' and a note: 'Accounts are the top-most level of organization and contain one or more tracking IDs.' The 'Setting up your property' section includes a 'Website Name' field with 'Lab 7', a 'Website URL' field with 'http://' and 'pgbovine-hci-lab7.herokuapp.com', and an 'Industry Category' dropdown menu set to 'Computers and Electronics'.

Get your tracking code & copy into your HTML/handlebars files, and paste right before `</body>` near the end of each file

The screenshot shows the Google Analytics Admin interface. At the top, there are navigation tabs: HOME, REPORTING, CUSTOMIZATION, and ADMIN (highlighted). Below the navigation, the breadcrumb path is Administration > HCI A/B Testing / Lab 7. On the left side, there is a sidebar with a 'PROPERTY' dropdown menu set to 'Lab 7'. Below the dropdown are links for Property Settings, User Management, and Tracking Info. The Tracking Info section is expanded, showing options like Tracking Code, Data Collection, User-ID, Session Settings, Organic Search Sources, Referral Exclusion List, and Search Term Exclusion List. The main content area displays the Tracking ID as UA-87137144-1 and the Status as 'No data received in past 48 hours. Learn more'. Below this, there is a section for 'Website tracking' with a brief explanation and instructions to copy the tracking code. The tracking code is provided in a code block within a light gray box.

HOME REPORTING CUSTOMIZATION **ADMIN**

Administration
HCI A/B Testing / Lab 7

PROPERTY
Lab 7

Property Settings

User Management

.js Tracking Info

- Tracking Code**
- Data Collection
- User-ID
- Session Settings
- Organic Search Sources
- Referral Exclusion List
- Search Term Exclusion List

Tracking ID
UA-87137144-1

Status
No data received in past 48 hours. [Learn more](#)

Website tracking

This is the Universal Analytics tracking code for this property.
To get all the benefits of Universal Analytics for this property, copy and paste this code into every webpage you want to track.

```
<script>
(function(i,s,o,g,r,a,m){i['GoogleAnalyticsObject']=r;i[r]=i[r]||function(){
(i[r].q=i[r].q||[]).push(arguments)},i[r].l=1*new Date();a=s.createElement(o),
m=s.getElementsByTagName(o)[0];a.async=1;a.src=g;m.parentNode.insertBefore(a,m)
})(window,document,'script','https://www.google-analytics.com/analytics.js','ga');

ga('create', 'UA-87137144-1', 'auto');
ga('send', 'pageview');

</script>
```

If you need to find the tracking code again


- Look in the Admin tab:

The screenshot shows a web analytics dashboard interface. At the top left, there is a logo and the text "Lab 7" and "All Web Site Data". Below this is a navigation bar with tabs: "HOME", "REPORTING", "CUSTOMIZATION", and "ADMIN" (which is highlighted with an orange underline). Under the "ADMIN" tab, the page title is "Administration" and "HCI A/B Testing". The main content area is split into two columns. The left column is titled "ACCOUNT" and contains a dropdown menu with "HCI A/B Testing" selected, and a list of options: "Account Settings", "User Management", "All Filters", "Change History", and "Trash Can". The right column is titled "PROPERTY" and contains a dropdown menu with "Lab 7" selected, and a list of options: "Property Settings", "User Management", ".js Tracking Info", "Tracking Code", "Data Collection", "User-ID", "Session Settings", "Organic Search Sources", "Referral Exclusion List", and "Search Term Exclusion List". A green rounded rectangle highlights the ".js Tracking Info" and "Tracking Code" options in the right column.

Test: is your webpage sending data to Google?

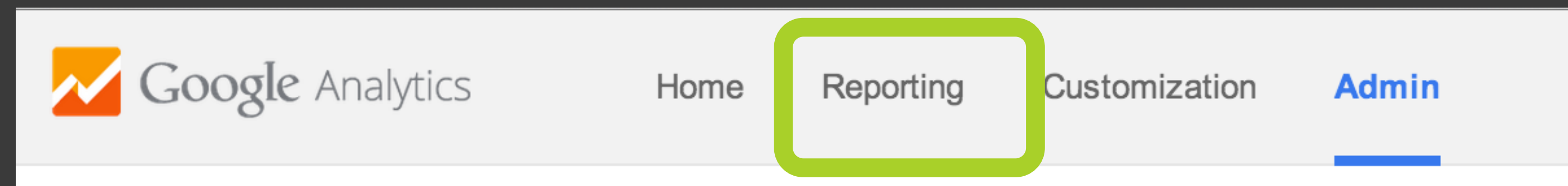
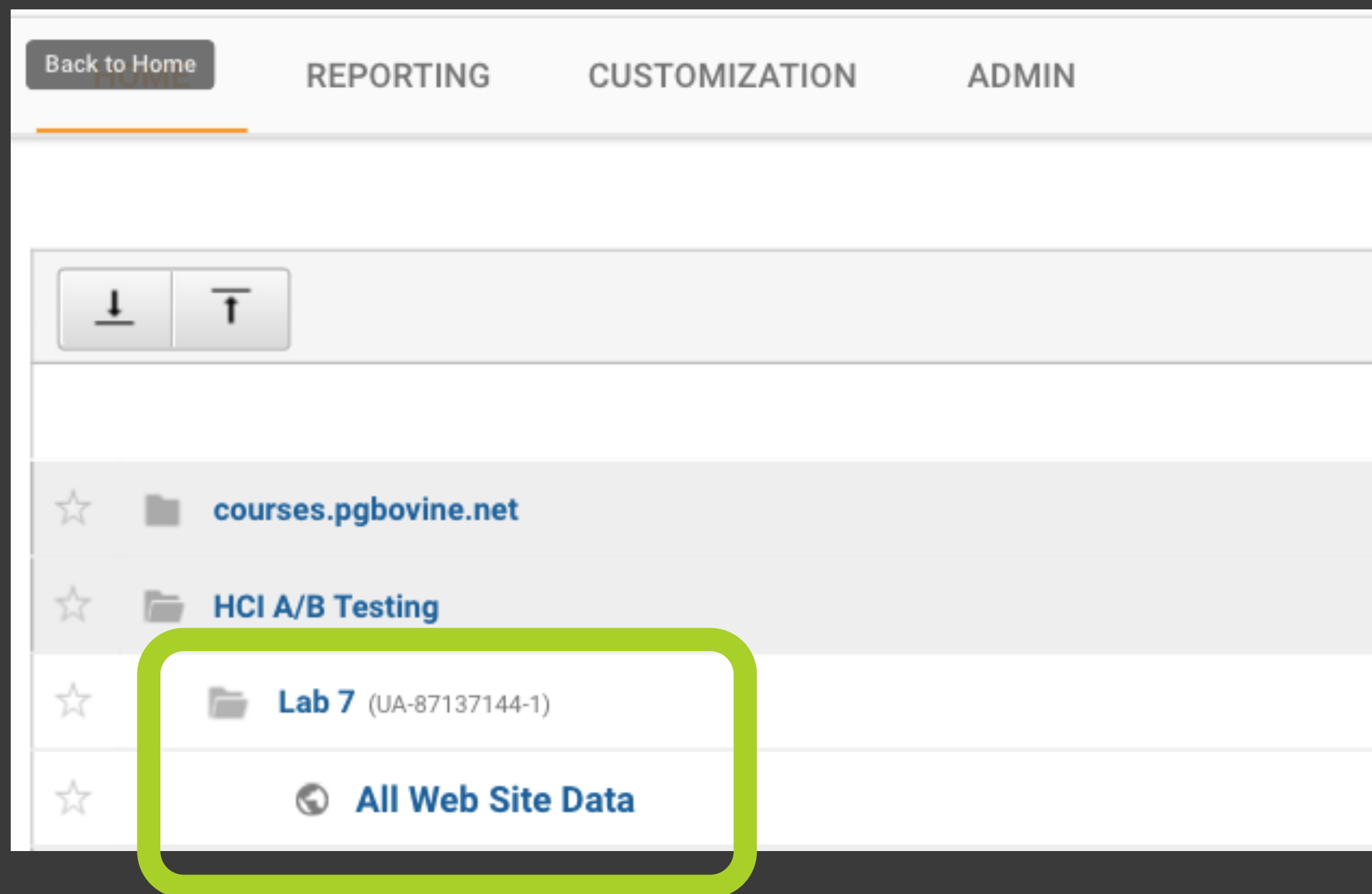
- Reload node.js, then open the Chrome developer tools at localhost:3000 and click to the “Network” tab.
- Reload the page in your browser.
- Do you see a ‘collect’ message initiated by ‘Other’ in the Network tab? (Hint: you might need to filter for ‘collect’...)

The screenshot shows the Chrome DevTools Network tab with a filter for 'collect'. The timeline shows a request starting at approximately 200ms and ending at 270ms. The request details table below is highlighted with a green border.

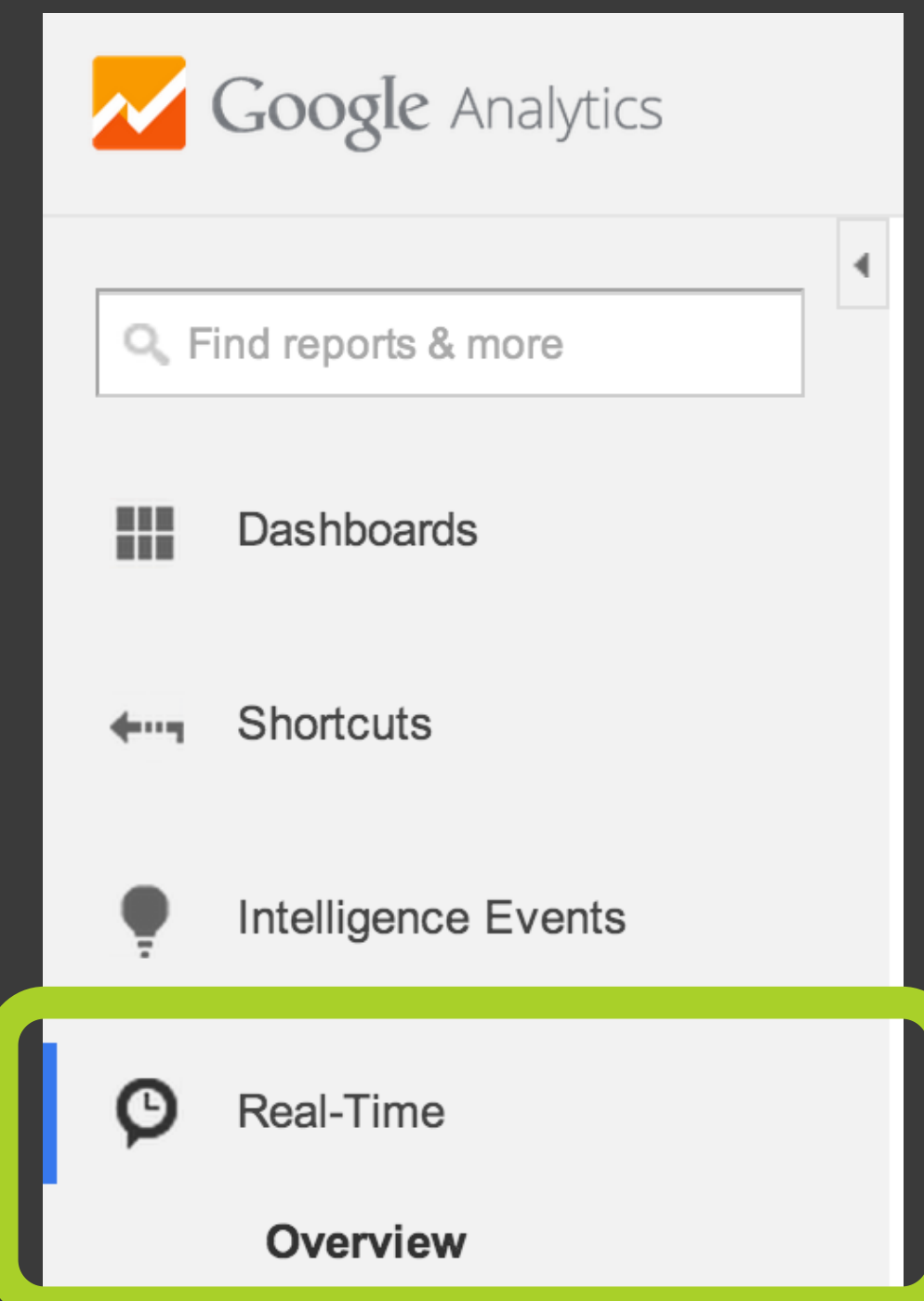
Name Path	Status Text	Type	Initiator	Size Content	Time Latency	Timeline - Start Time	600.00 ms	800.00 ms	1.00 s
 collect?v=1&_v=j41&a=... www.google-analytics....	200 OK	gif	Other	387 B 35 B	28 ms 27 ms				

Test: is it receiving data? (part I)

- Find “All Web Site Data”, click on the “Reporting” tab



- Click “Real-Time”, then “Overview”



Test: is it receiving data? (part 2)

- Reload <http://localhost:3000> and wait five seconds.
- How many Active Visitors does Google Analytics think you have on the site? (Hopefully one.)

Right now

1

active visitors on site

DESKTOP

100%

Pageviews

Per minute

1.5

1.0

0.5

-26 min

-21 min

-16 min

-11 min

-6 min

...

-1

Per second

1.5

1

0.5

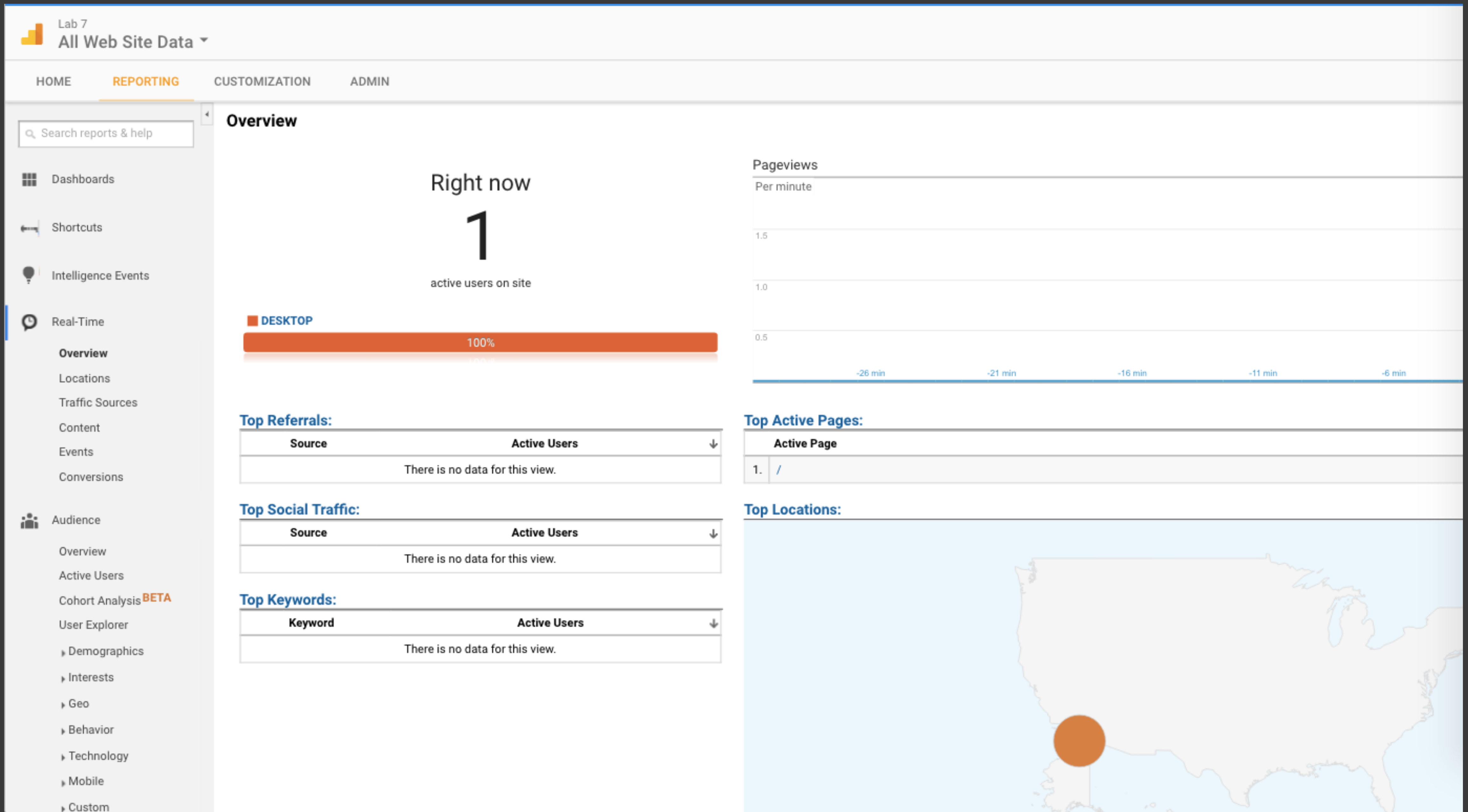
-60 sec

-45 sec

-30 sec

-15 sec

Test: is it receiving data? one visitor from San Diego!



Visitor patterns over time

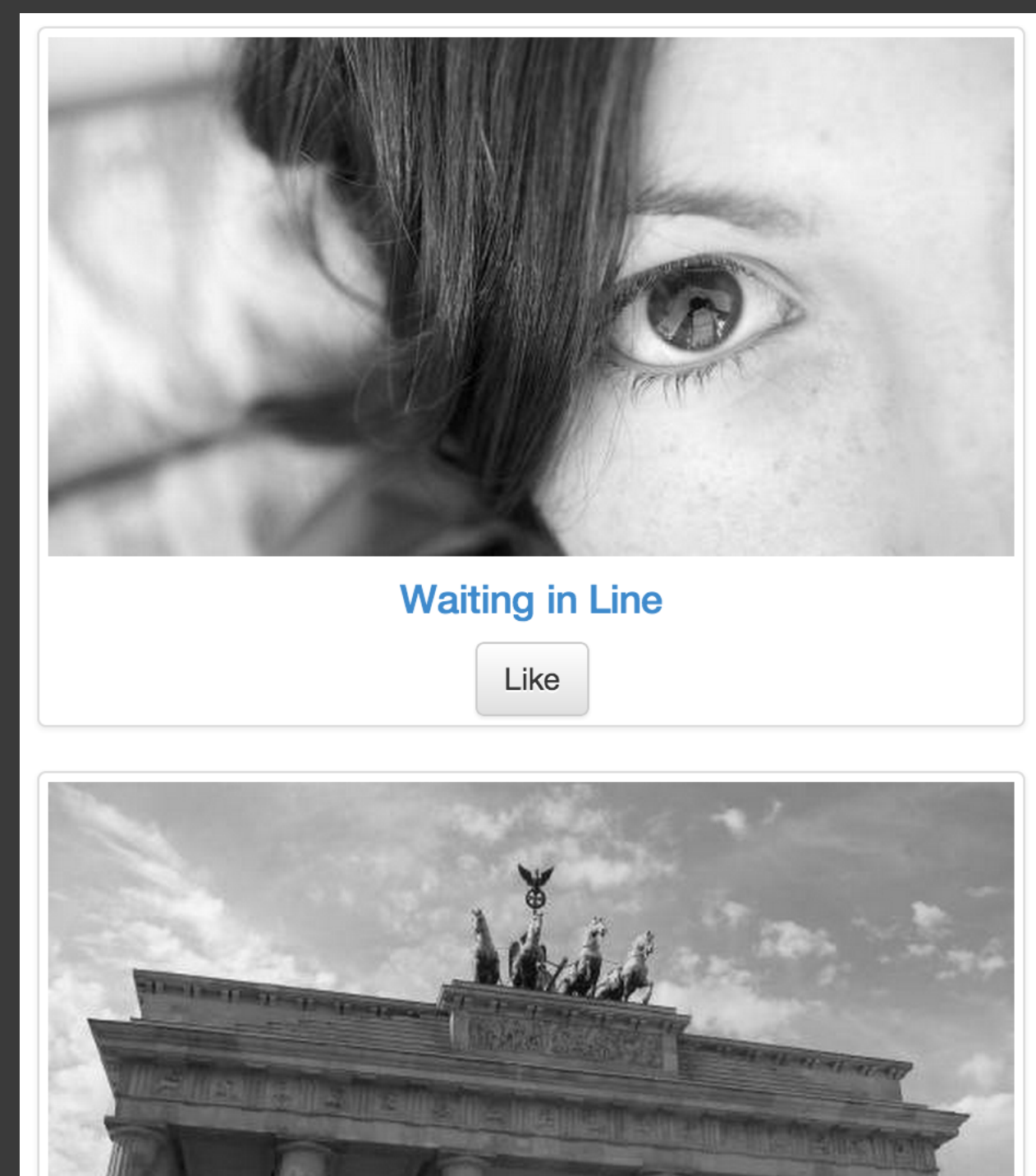
- Click on the Audience tab and then Overview to see how many people looked at your site, and which pages they saw.
- This will be empty right now; it typically gets updated each night.
- This will be useful for tracking your online user studies!

practice

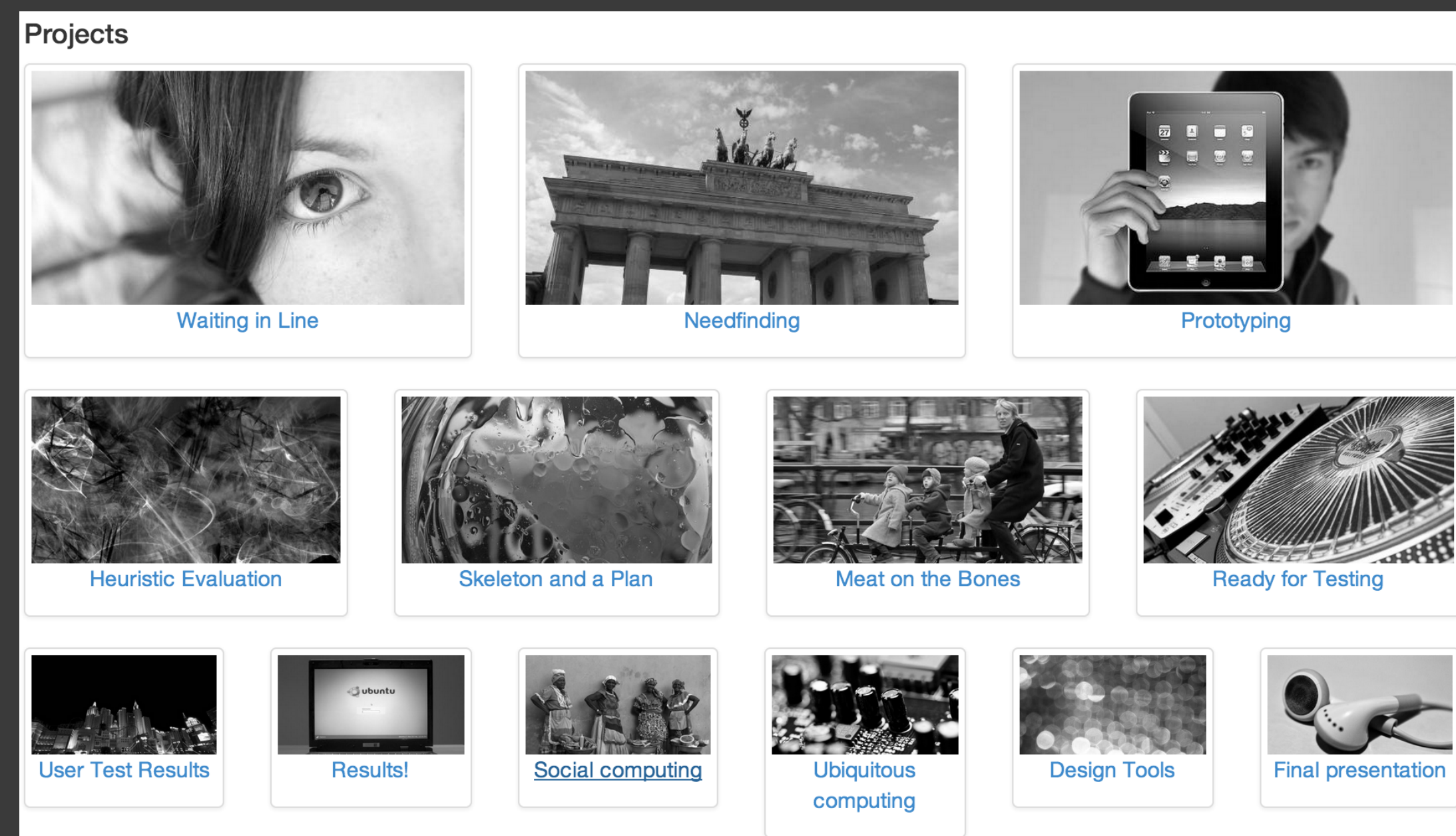
A/B testing

Goal: run an A/B test on a new grid layout vs. the original vertical layout

Which layout gets users to click “Like” on more projects?



Original Vertical Layout



New Grid Layout (we will fake it for this lab)

Register the control and experimental URL routes

- What do I do?
 - Register a new route to `/grid` that routes to a new controller in `index.js` called `index.viewGrid`.
- How do I do it?
 - Look in `app.js`. You already have a route for `/` (which means localhost: 3000) registered to `index.view`. Create another route to `/grid` that connects to `index.viewGrid`.
- How do I know it's working?
 - You won't be able to check this until you create the controller. If you try to start `node.js` now, it will complain.

Create `index.viewGrid`

- What do I do?
 - In `routes/index.js`, create a new controller that exports the function `viewGrid`. It should also render `index.handlebars`.
- How do I do it?
 - Your controller should be identical to `index.view` for now, just with a different name.
- How do I know it's working?
 - Reload `node.js` and open `localhost:3000` and `localhost:3000/grid`. Do they both render the same page? If yes, then you're good for now.

Send a boolean flag to the template

- What do I do?
 - Send a boolean variable called `grid` to the handlebars template so that we can use an if/else block to render different versions.
- How do I do it?
 - In `index.view`, add a property 'grid' to the projects JSON and set it to `false`, e.g. `projects["grid"] = false;`
 - In `index.viewGrid`, add a property 'grid' to the projects JSON and set it to `true`.
 - Now, when projects is sent to the Handlebars template, you'll have either `grid = false` or `grid = true`.
- How do I know if it's working? (You can't, yet.)

If/else block to render a (fake) grid

- What do I do?

- In `index.handlebars`, render the HTML "grid! <button class="likeBtn">Like</button>" if `grid` is `true`. Keep the original `{{#each projects}}` block if `grid` is `false`

- How do I do it?

- In Handlebars, `{{#if myVar}}` will test if `myVar` is `true`

- `{{#if myVar}}`
...code...
`{{else}}`
...code...
`{{/if}}`

- How do I know if it's working? `http://localhost:3000/grid` shows "grid! <Like button>"; `http://localhost:3000` shows the old site

Record an event whenever the user clicks a Like button on a project

- What do I do?
 - We will be measuring which layout produces more “Like” clicks.
 - In `public/js/introHCI.js`, record a Google Analytics event whenever the user clicks *any* Like button. Use Category: ‘like’, Action: ‘click’.
 - Remember the category and action; you’ll need them later!
- How do I do it?
 - You’ll need to register a click handler
 - The click handler should send an Analytics event
- How do I know if it’s working? (Go to the next slide.)

Test: is the event sending data?

- Open the Chrome developer tools at localhost:3000 and click to the “Network” tab.
- Reload the page and click on a “Like” button for a project.
- It should trigger another network event to ‘collect’.

The screenshot shows the Chrome DevTools Network tab. The search bar contains the text 'collect'. The filter is set to 'All'. The network log shows two entries for 'collect?v=1&_v=j41&a=6... www.google-analytics.com'. The second entry is highlighted with a green box. The timeline shows a red vertical line at approximately 900,000 ms, corresponding to the highlighted event. The table below the timeline provides details for the highlighted event.

Name Path	Status Text	Type	Initiator	Size Content	Time Latency	Timeline - Start Time
collect?v=1&_v=j41&a=6... www.google-analytics.com	200 OK	gif	Other	386 B 35 B	23 ms 23 ms	
collect?v=1&_v=j41&a=6... www.google-analytics.com	200 OK	gif	Other	386 B 35 B	23 ms 22 ms	

Test: Google Analytics receiving data

- Just like before, go to “Real-Time”, but this time click on “Events” to see a realtime update with the events.

The screenshot shows the Google Analytics interface for 'Lab 7' under 'All Web Site Data'. The 'REPORTING' tab is selected, and the 'Events' section is active. The main display shows 'Right now 1 active users on site'. A bar chart titled 'Events Per minute' shows a single bar at -1 min with a value of 1. Below the chart, a table lists the event details.

Viewing: **Active Users** Events (Last 30 min)

Active Users with Events: **1 (100% of total)**

Event Category	Event Action
1. like	click

Before setting up the Google Analytics experiment, you'll need to publish the current version of your site to Heroku so that Google can see your A/B alternatives.

Deploy your lab7 to Heroku:

- Create a new application in Heroku and link to GitHub
- Mine will be called: <http://pgbovine-hci-lab7.herokuapp.com>

The screenshot shows the Heroku dashboard for the application 'pgbovine-hci-lab7'. At the top, there is a search bar and navigation links for 'Personal apps' and 'pgbovine-hci-lab7'. Below this, there are tabs for 'Overview', 'Resources', 'Deploy', 'Metrics', 'Activity', 'Access', and 'Settings'. The main content area is divided into two sections. The first section, 'Add this app to a pipeline', includes a sub-section 'Add this app to a pipeline' with instructions to create a new pipeline or choose an existing one. It also features two informational cards: one explaining that pipelines connect multiple apps and promote code, and another stating that pipelines connected to GitHub can enable review apps. Below these cards are buttons for 'New Pipeline...' and 'Add to a Pipeline'. The second section, 'Deployment method', shows three options: 'Heroku Git' (Use Heroku CLI), 'GitHub' (Connected), and 'Dropbox' (Connect to Dropbox). The 'GitHub' option is highlighted with a green checkmark. At the bottom, there is a section 'App connected to GitHub' showing the connection to 'pgbovine/lab7' with a 'Disconnect...' button and a note that releases in the activity feed link to GitHub to view commit diffs.

Deploy your lab7 to Heroku:

- Deploy “master” branch to Heroku and visit both the original and grid URLs to test:
 - Mine will be called: <http://pgbovine-hci-lab7.herokuapp.com>
 - Grid version: <http://pgbovine-hci-lab7.herokuapp.com/grid>

The screenshot shows the Heroku deployment interface. On the left, under "Manual deploy", it says "Deploy the current state of a branch to this app." The main section is titled "Deploy a GitHub branch" and includes a dropdown menu with "master" selected and a "Deploy Branch" button. Below this, three steps are listed with green checkmarks: "Receive code from GitHub", "Build master" (with a "Show build log" link), and "Deploy to Heroku". A message at the bottom states "Your app was successfully deployed." with a "View" button.

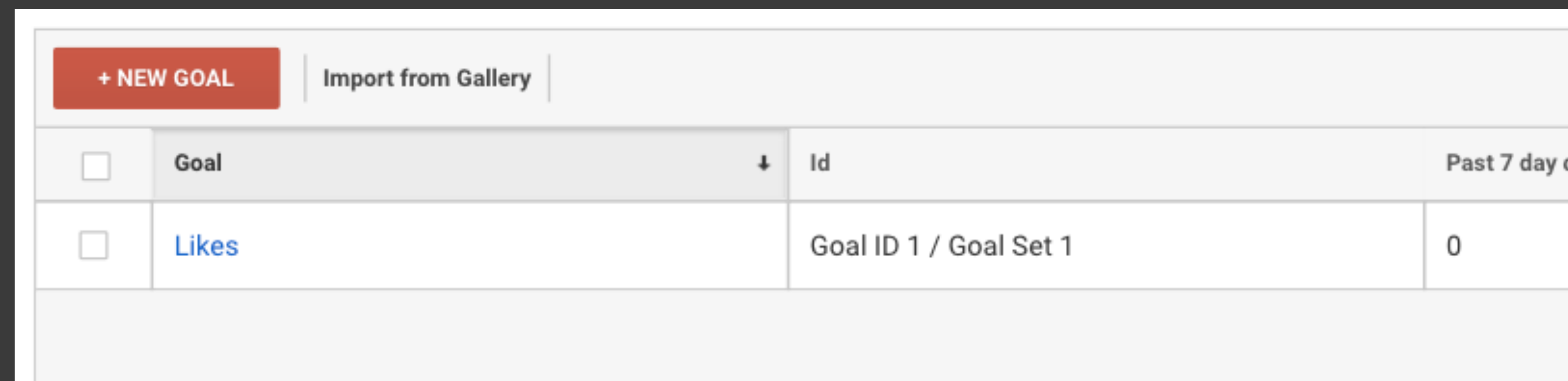
Randomizing & Logging A/B Experiments

Google Analytics Experiments

- The way Google Analytics Experiments works is that, once set up, it will automatically 'coin flip' and choose a random A/B for a user
- It will also ensure that when that user returns to the page, the same A/B version that was randomly selected will be shown again

Add your custom 'like' event as a goal

- Click on the “Admin” tab at the top of Google Analytics
- Click on “Goals” in the right column, then “+New Goal”
- Under “Goal Setup”, choose “Custom”. Click “Continue”.
 - Name: “Likes”
 - Type: “Event”
- Recall our event description from before:
 - category equals “like”
 - action equals “click”
- Click “Create Goal”
- Back on starter page, see:

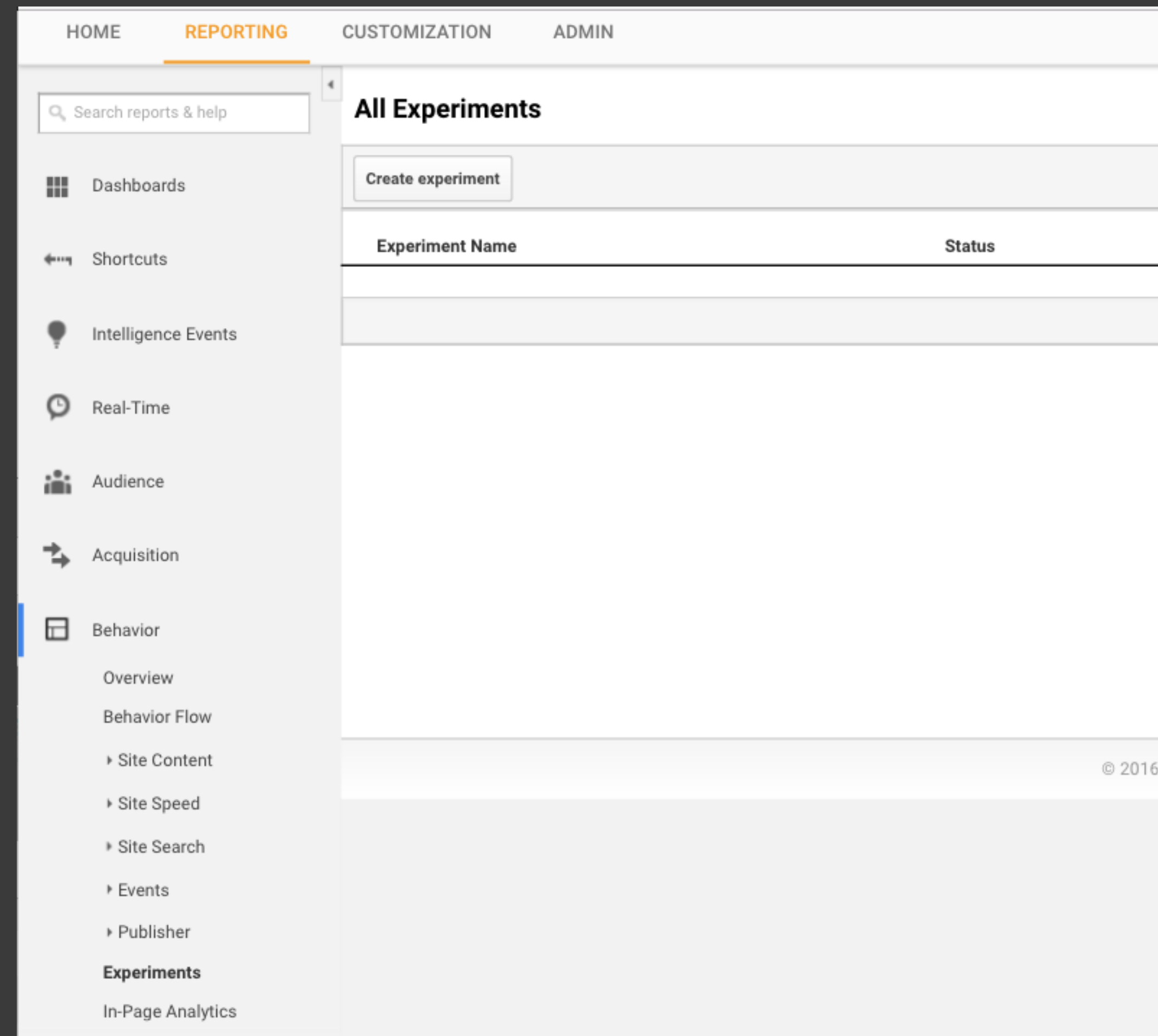
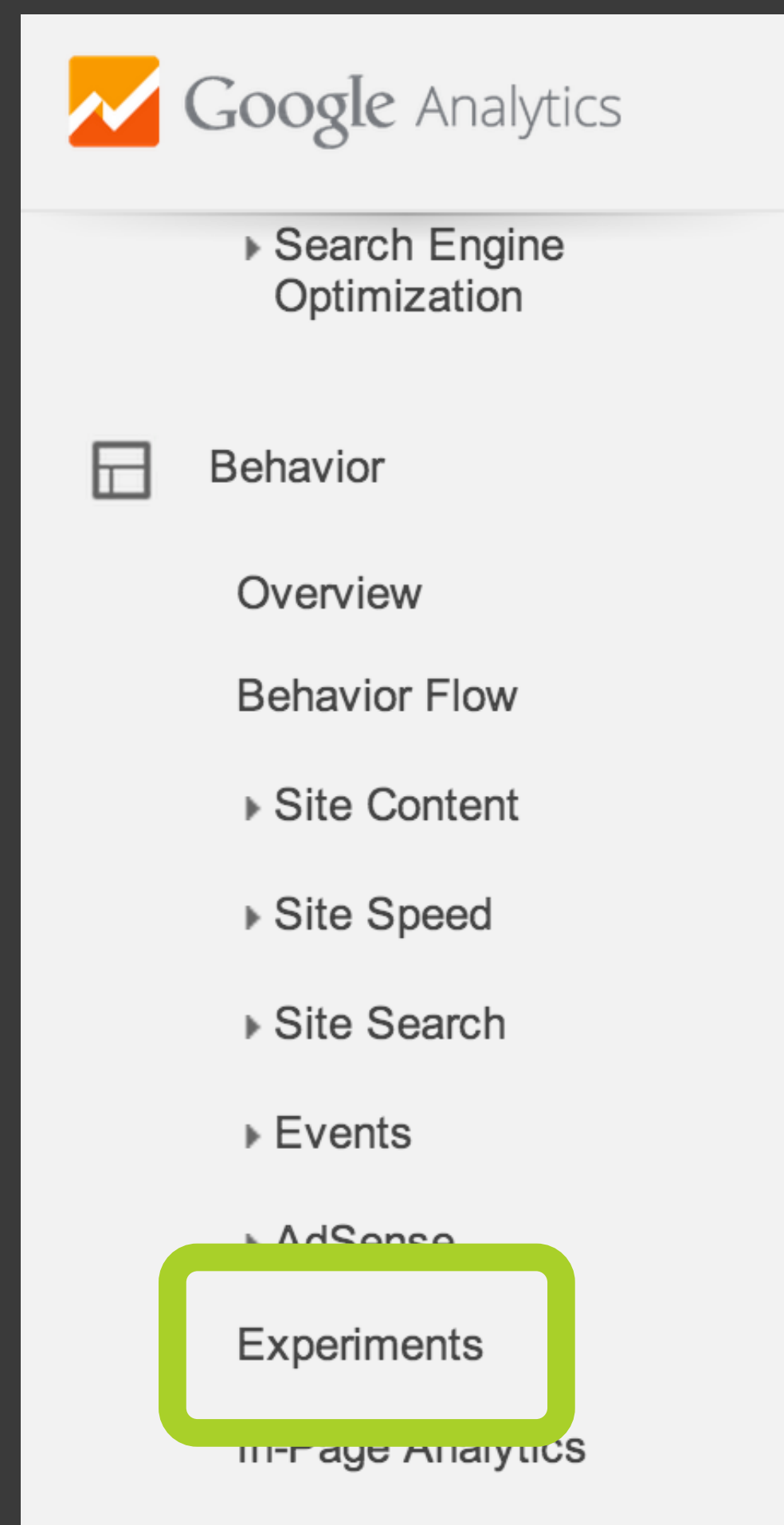


<input type="checkbox"/>	Goal	Id	Past 7 day
<input type="checkbox"/>	Likes	Goal ID 1 / Goal Set 1	0

19 goals left

Create a Google Analytics experiment

- In the Google Analytics window, click back on the Reporting top-level tab. Then click on the Behavior side-tab and finally Experiments



Choose your experiment objective

- Enter the URL of the homepage
e.g., <http://pgbovine-hci-lab7.herokuapp.com/>
- Name the experiment (e.g., “Grid Likes”)
- Click on the “Select a Metric” dropdown and look within Goal Set 1 to find the goal we just created
- Click on “Advanced options” and set the minimum time to 3 days
- Click “Next Step”

Content Experiments - Create a new experiment

1 Choose an experiment objective

Name for this experiment

Grid Likes

Objective for this experiment ?

Likes (Goal 1 Completions) × - or - [Create a new object](#)

Percentage of traffic to experiment ?

100% ▾

Email notification for important changes

OFF

[Advanced Options](#) ?

[Next Step](#) [Save for Later](#) [Discard](#)

Enter the URLs of both your page versions

- The web page to experiment should preview your default page. Add the alternate grid as Variant 1 ...
- You can name the page as you like, e.g., “Grid”
- Click “Next Step”

Content Experiments - Create a new experiment

edit

✓ Choose an experiment objective

The experiment, Grid Likes, will evaluate Likes (Goal 1 Completions) on 100% of user traffic. Email notification is not set for this experiment. 3 days minimum run time, 95% confidence threshold.

2 Configure your experiment

Original Page URLs with dynamic parameters are supported. ⓘ

Web page to experiment ⓘ

http:// pgbovine-hci-lab7.herokuapp.com/

Name for the page

Original

Consolidate experiment for other content reports ⓘ

Variant 1 ⓘ

Web page to experiment ⓘ


http:// pgbovine-hci-lab7.herokuapp.com/grid

Name for the page

Grid

Michael Bernstein
human-computer interaction · social computing · crowdsourcing

Projects



Michael Bernstein
human-computer interaction · social computing · crowdsourcing

Projects

grid Like

Insert the experiment JavaScript

- Choose “Manually insert the code”
 - This gives you JavaScript code that will run your experiment
- Put that code inside of your views/index.handlebars file *near the very top right after the <head> tag (putting it at this location is very important!!!)*

3 Setting up your experiment code

How do you want to set up your experiment code?

Manually insert the code Send the code to webmaster

Adding script code to your page ?

1. Make sure your original and variant pages have [Google Analytics tracking code installed](#).
2. Then, paste this experiment code immediately after the opening head tag at **the top** of your original page.

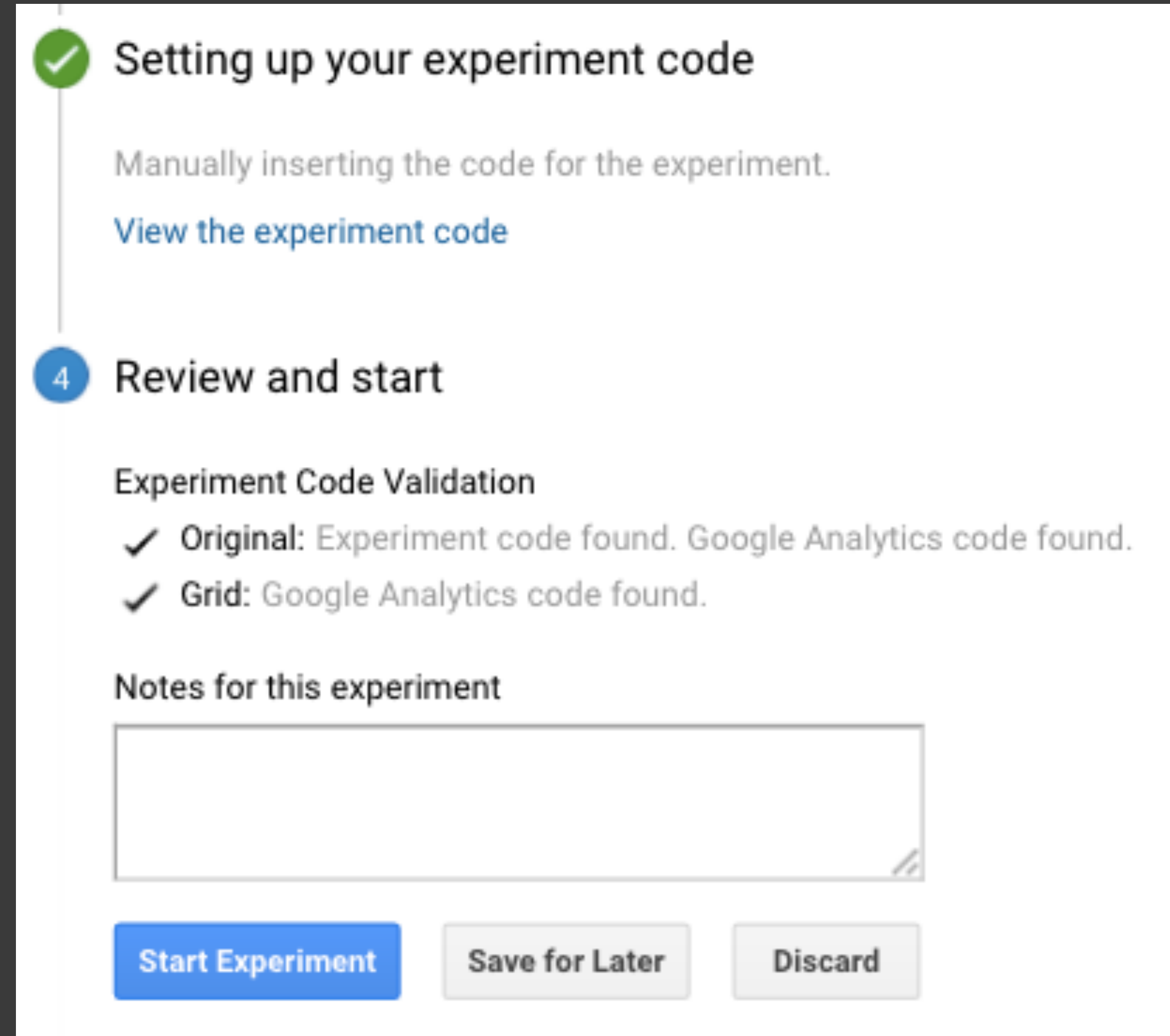
```
<!-- Google Analytics Content Experiment code -->
<script>function utmx_section(){function utmx(){(function(){var
k='133364162-0',d=document,l=d.location,c=d.cookie;
if(l.search.indexOf('utm_expid='+k)>0)return;
function f(n){if(c){var i=c.indexOf(n+'=');if(i>-1){var j=c.
indexOf(';');return escape(c.substring(i+n.length+1,j<0?c.
length:j))}}var x=f('__utmx'),xx=f('__utmxx'),h=l.hash;d.write(
'<sc'+ript src='"+http'+(l.protocol=='https:'?'s://ssl':
'://www')+'.google-analytics.com/ga_exp.js?'+'utmkey='+k+
'&utmx='+x?'x:')+'&utmxx='+xx?'xx:')+'&utmtime='+new Date().
valueOf()+h?'&utmhash='+escape(h.substr(1)):'')+
" type="text/javascript" charset="utf-8"></sc'+ript>');
</script><script>utmx('url','A/B');</script>
<!-- End of Google Analytics Content Experiment code -->
```

Push the experiment JavaScript to Heroku

- Git add all the new/changed files
- Git commit your new code and push the update to GitHub and deploy again on Heroku
- Click “Next Step” in Google Analytics after you’ve updated on Heroku

Validate your experiment setup

- Google will now search to see if you have correctly included the experiment code on your page.
 - If it can't find the code, make sure you pushed it correctly to Heroku. Then click “Revalidate”
- If it works, click “Start Experiment”



The screenshot shows the 'Review and start' step of the experiment setup process. It features a progress indicator with a green checkmark for the previous step and a blue circle with the number '4' for the current step. The current step is titled 'Review and start' and includes a section for 'Experiment Code Validation' with two checked items: 'Original: Experiment code found. Google Analytics code found.' and 'Grid: Google Analytics code found.'. Below this is a text area for 'Notes for this experiment' and three buttons: 'Start Experiment' (highlighted in blue), 'Save for Later', and 'Discard'.

Setting up your experiment code

Manually inserting the code for the experiment.

[View the experiment code](#)

4 Review and start

Experiment Code Validation

- ✓ Original: Experiment code found. Google Analytics code found.
- ✓ Grid: Google Analytics code found.

Notes for this experiment

[Start Experiment](#) [Save for Later](#) [Discard](#)

Is it working?

- If you visit <http://pgbovine-hci-lab7.herokuapp.com/> (or whatever you named your version), you will randomly either see the original or grid layouts
- For the moment, you'll have to look at the Real-Time view in Google Analytics to make sure you're receiving pageviews and Like click events
- The experiment page will update only after ~24 hours, so **don't wait until the last minute to do your assignment**
- You'll always see the same A/B version that was randomly selected the first time ... if you would like Google Analytics to serve a new random page to you, visit it in an Incognito/Private tab in your browser (or clear all of your cookies)