

announcements 07-Sep-2017

join [#jobs-and-internships](#) on Slack

announcements 07-Sep-2017

join #jobs-and-internships on Slack

a1 grades ready; coming soon...

a2 due tonight; a3 out today

announcements 07-Sep-2017

vote!

poll on whether we do catering or not



help wanted: final demo day organization

do you...

know how to book rooms?

know how to make web visualizations?
(funding page)

have a car to pick stuff up?



tasty sandwiches? I'm in

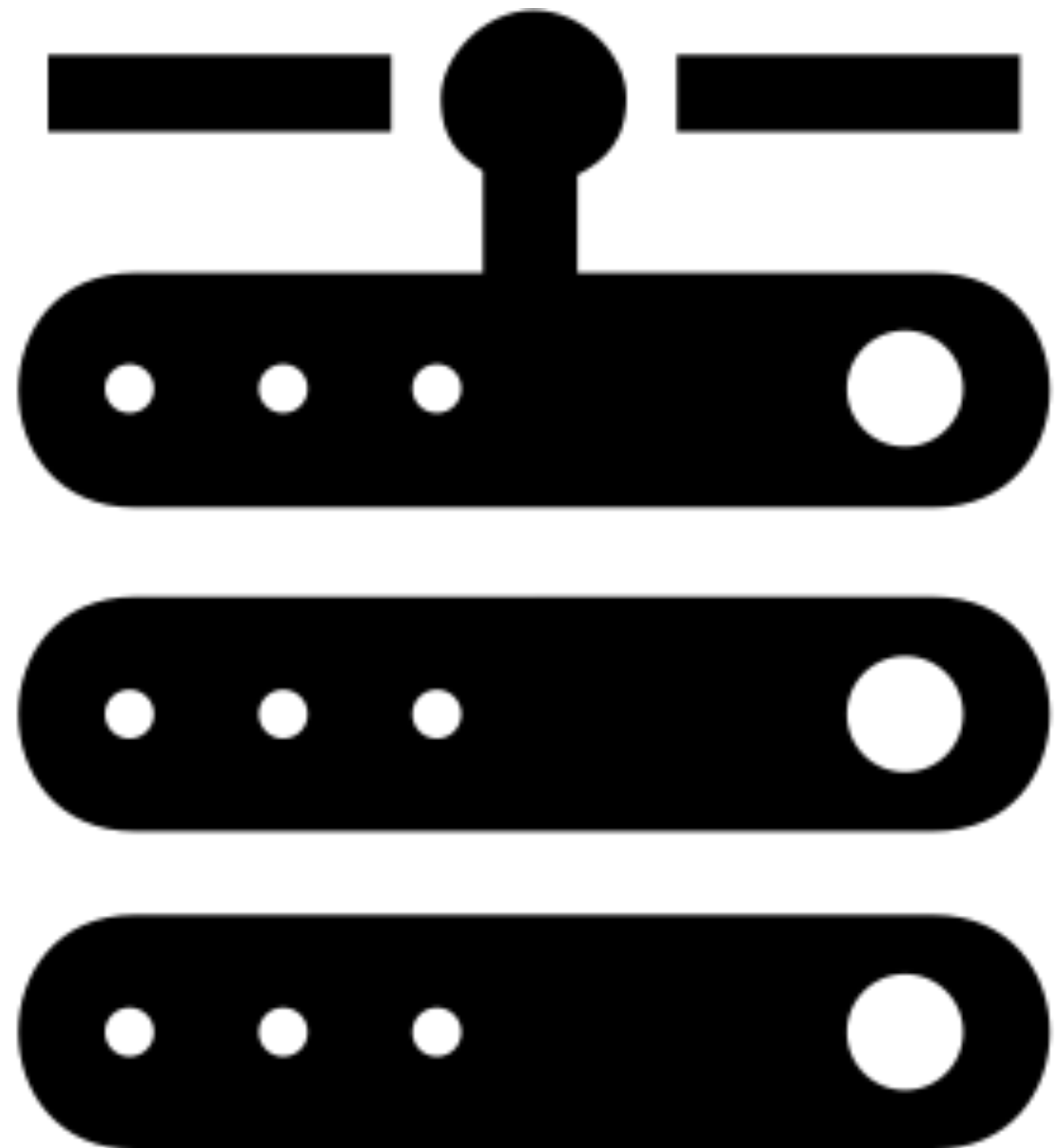
msg me on slack w/ what you can help
with. I'll add to our channel



CS4241

WebWare

servers, http, forms

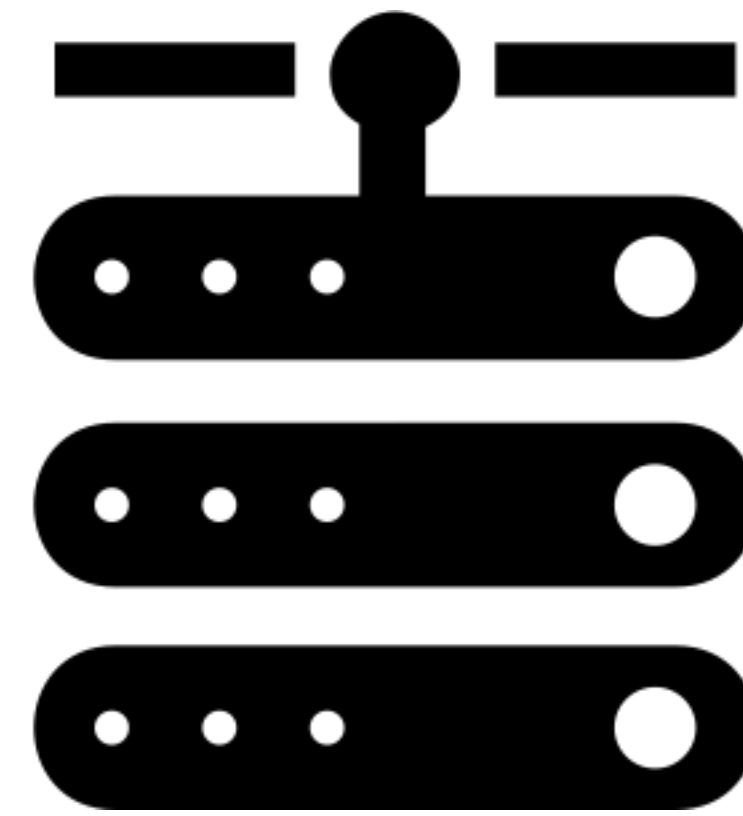


servers

browsers



servers



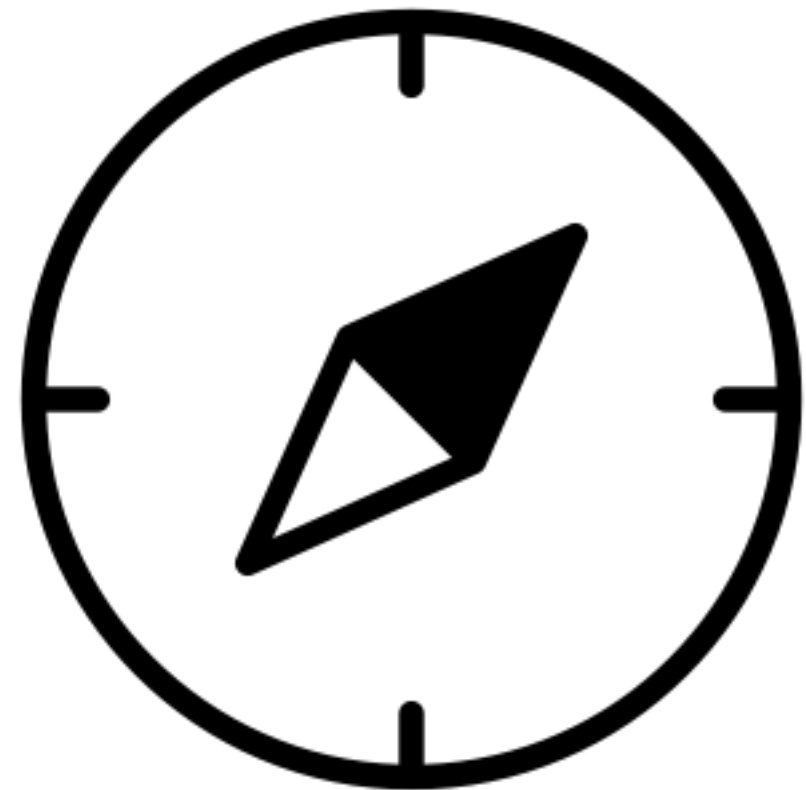
db



http



local
(net)



front-end

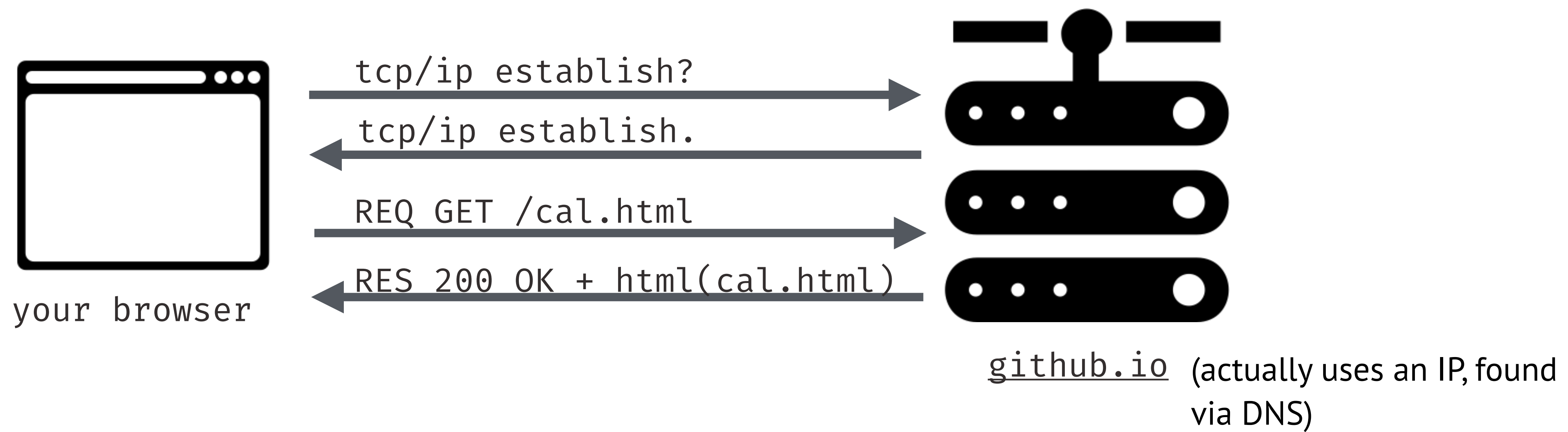
back-end

given a url:

<https://cs4241-17a.github.io/cal.html>

browser gets HTML from webserver

GET implies a protocol, in this case, HTTP



requesting cal.html

pretend we're using http

```
header | GET /cal.html HTTP/1.1
        | Host: cs4241.github.io
        | User-Agent: Chrome:60.0.3112.113
        | Accept: text/html, */*
        | Accept-Language: en-us
        |
        | # body goes after blank line
```

let's take a look...

http methods (think VERB)

it's what you *do*

GET - get a URL

HEAD - get info about a URL

PUT - create or update something at a URL

POST - send data to URL

DELETE - delete a URL (or something referenced by it)

GET AND POST for now...

response from github.io

REMEMBER: It's just Text...

header

```
HTTP/1.1 200 OK
Date: Thu, 07 Sep 2017 12:38:32 GMT
Server: github.com
content-length: 2721
content-type: text/html; charset=utf-8
```

body

```
<!DOCTYPE html>
<html lang="en">

  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, height=device-height">
```


http status codes

200 - OK

404 - Not Found

503 - Service Unavailable

500 - Internal Server Error

403 - Forbidden

and many more



basic server:
request ->
response

```
var http = require('http')
    , fs   = require('fs')
    , url  = require('url')
    , port = 8080;

var server = http.createServer (function (req, res) {
    var uri = url.parse(req.url)
```

```
        default:
            res.end('404 not found')
    }
})
```

basic file server:
request ->
file ->
response

```
var http = require('http')
    , fs   = require('fs')
    , url  = require('url')
    , port = 8080;

var server = http.createServer (function (req, res) {
    var uri = url.parse(req.url)
```

```
function sendFile(res, filename, contentType) {
    contentType = contentType || 'text/html';

    fs.readFile(filename, function(error, content) {
        res.writeHead(200, {'Content-type': contentType})
        res.end(content, 'utf-8')
    })
}
```


better server:
request ->
code -> file ->
response



CGI

Common

Gateway

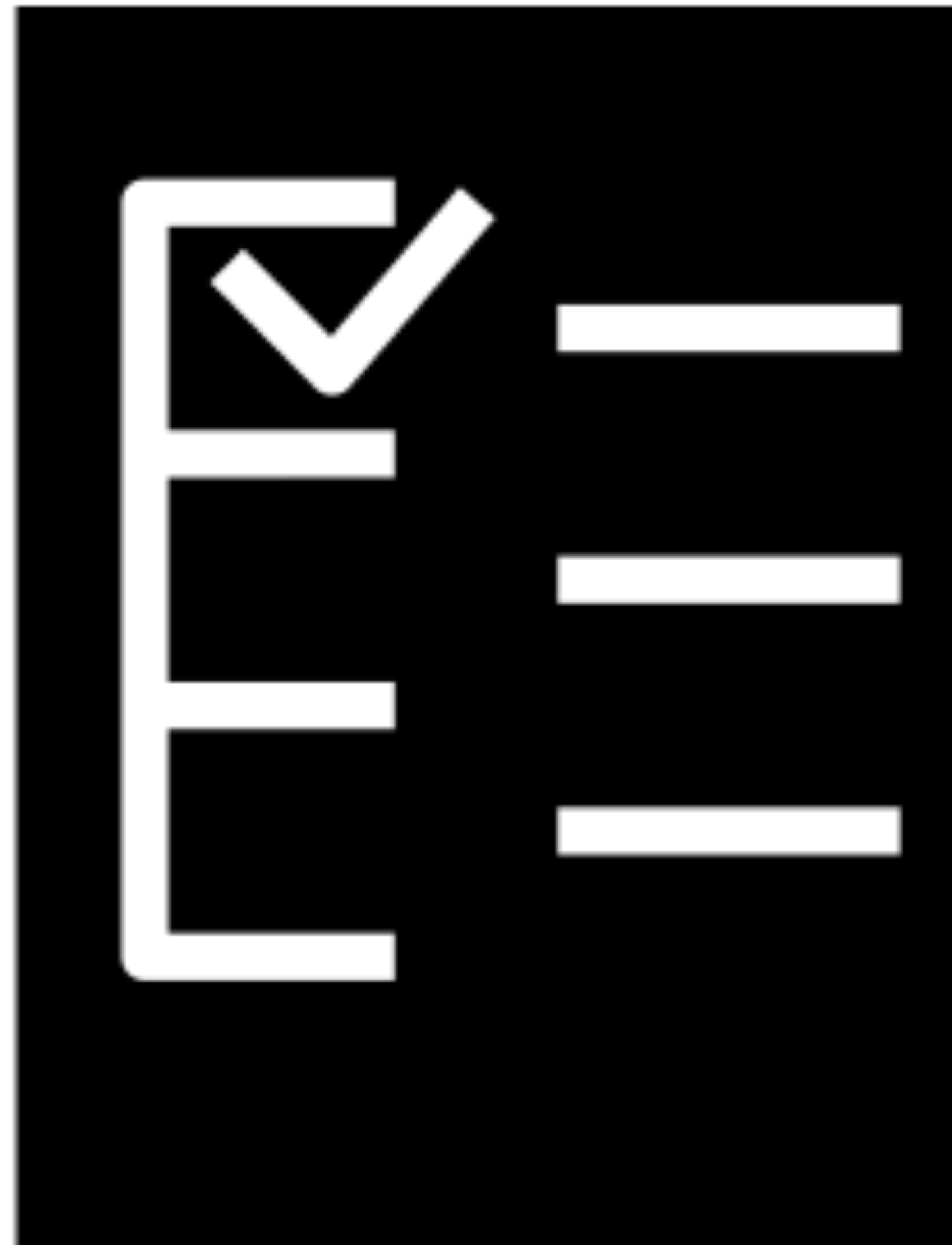
Interface

Dynamic Content is the point of CGI

CGI-bin built in to many servers

CGI example
bash

CGI example
C



html forms

live