

Going Serverless with VueJS

Divya Sasidharan



HELLO

MY NAME IS

Divya Sasidharan

HELLO

MY NAME IS

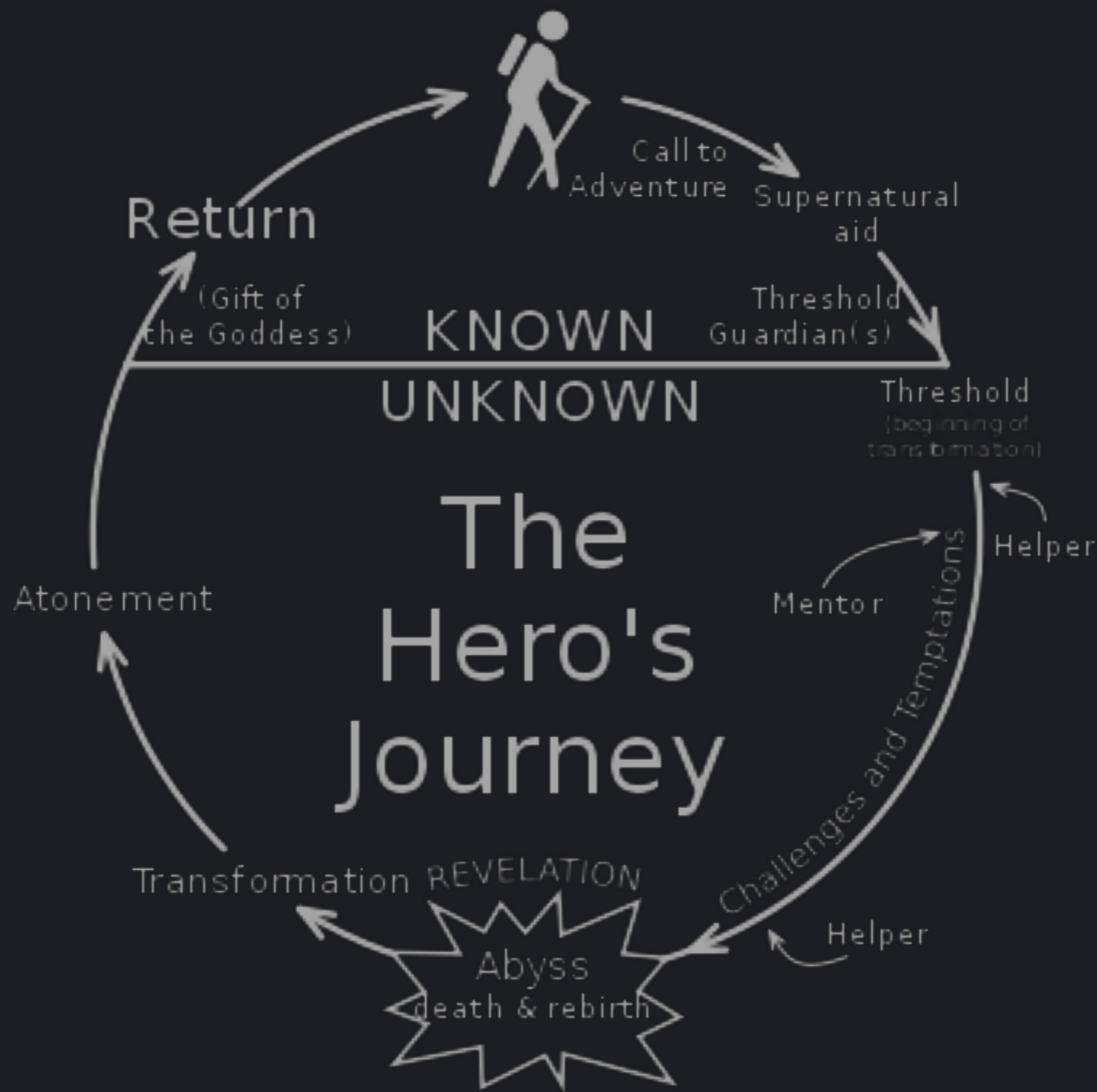
Divya Sasidharan

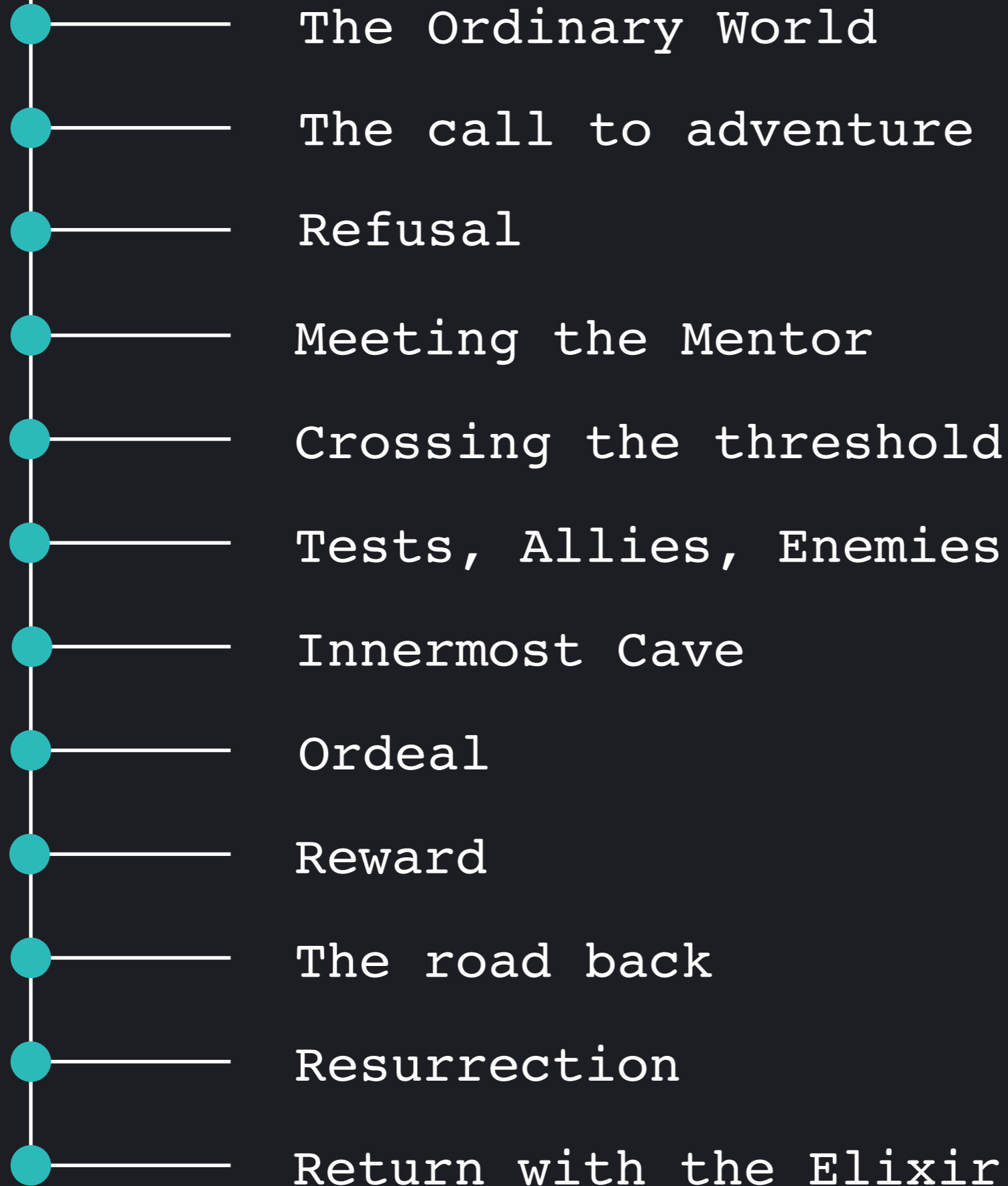
@shortdiv



netlify

THE HERO'S JOURNEY





-
- The Ordinary World
 - The call to adventure
 - Refusal
 - Meeting the Mentor
 - Crossing the threshold
 - Tests, Allies, Enemies
 - Innermost Cave
 - Ordeal
 - Reward
 - The road back
 - Resurrection
 - Return with the Elixir

DEPARTURE

- The Ordinary World
- The call to adventure
- Refusal
- Meeting the Mentor
- Crossing the threshold
- Tests, Allies, Enemies
- Innermost Cave
- Ordeal
- Reward
- The road back
- Resurrection
- Return with the Elixir

DEPARTURE

INITIATION

- The Ordinary World
- The call to adventure
- Refusal
- Meeting the Mentor
- Crossing the threshold
- Tests, Allies, Enemies
- Innermost Cave
- Ordeal
- Reward
- The road back
- Resurrection
- Return with the Elixir

DEPARTURE

INITIATION

RETURN

Act 1

Departure



The Ordinary World

DATE

SCENE

ACT

04/30

1

1



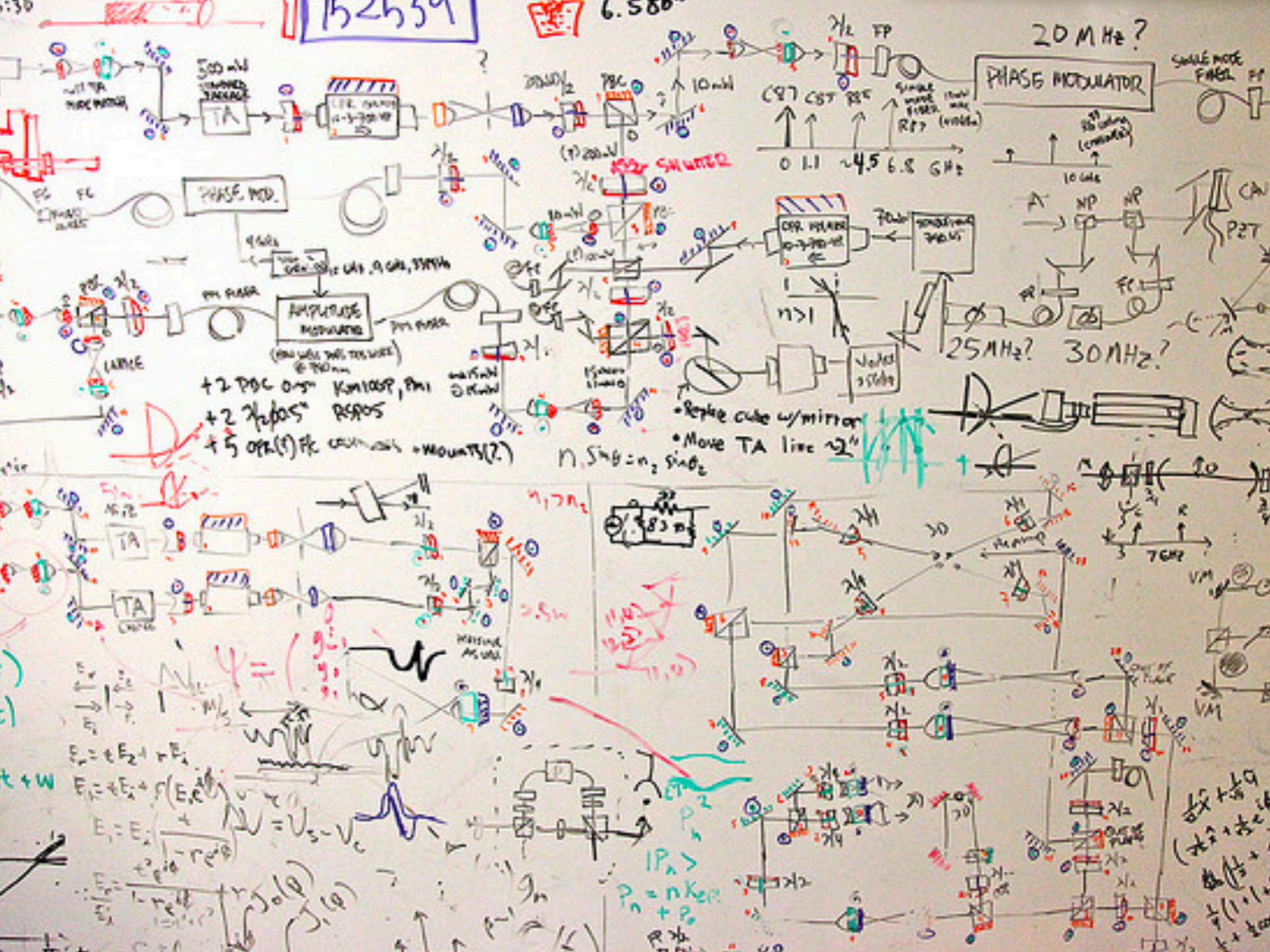
The Ordinary World

DATE	SCENE	ACT
04/30	1	1





Sammy Serverus





Lou Malnati's Pizzeria



5411 Reviews on Yelp

My Rating:



[Directions](#)

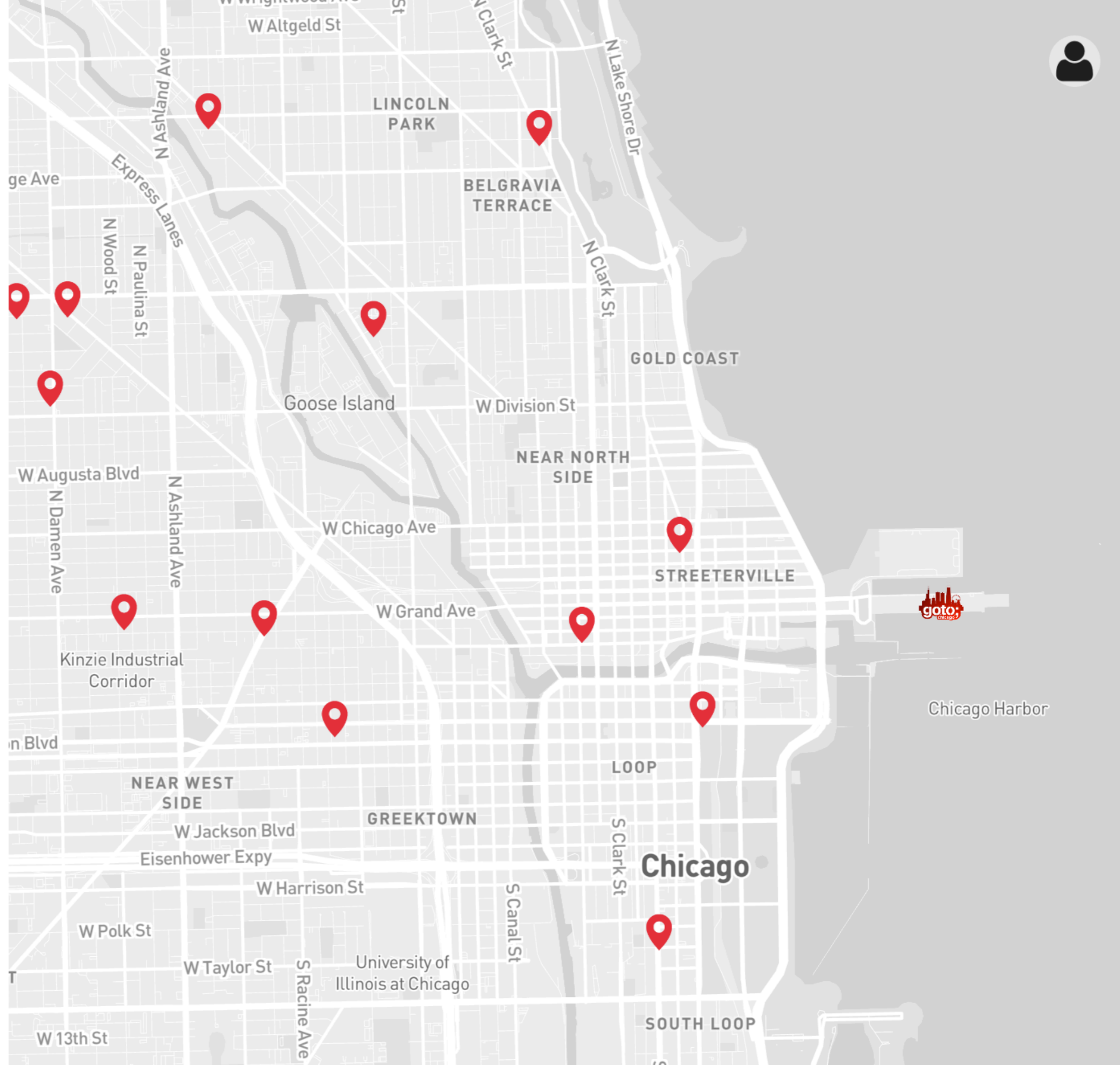


Pequod's Pizzeria



5324 Reviews on Yelp

My Rating:







Call to Adventure

DATE

SCENE

ACT

04/30

2

1



Call to Adventure

DATE

SCENE

ACT

04/30

2

1



Call to Adventure

DATE

SCENE

ACT

04/30

2

1



- 1. ▲ Some news (github.com)**
100000 points by somedude 3 hours ago | hide | 100000 comments
- 2. Some news (github.com)**
100000 points by somedude 3 hours ago | hide | 1 comments
- 3. Some news (github.com)**
100000 points by somedude 3 hours ago | hide | 100000 comments
- 4. Some news (github.com)**
100000 points by somedude 3 hours ago | hide | 100000 comments
- 5. Some news (github.com)**
100000 points by somedude 3 hours ago | hide | 100000 comments
- 6. Some news (github.com)**
100000 points by somedude 3 hours ago | hide | 100000 comments
- 7. Some news (github.com)**
100000 points by somedude 3 hours ago | hide | 100000 comments
- 8. Some news (github.com)**
100000 points by somedude 3 hours ago | hide | 100000 comments
- 9. Some news (github.com)**
100000 points by somedude 3 hours ago | hide | 100000 comments
- 10. Some news (github.com)**
100000 points by somedude 3 hours ago | hide | 100000 comments
- 11. Some news (github.com)**
100000 points by somedude 3 hours ago | hide | 100000 comments
- 12. Some news (github.com)**
100000 points by somedude 3 hours ago | hide | 100000 comments
- 13. Some news (github.com)**
100000 points by somedude 3 hours ago | hide | 100000 comments
- 14. Some news (github.com)**
100000 points by somedude 3 hours ago | hide | 100000 comments
- 15. Some news (github.com)**



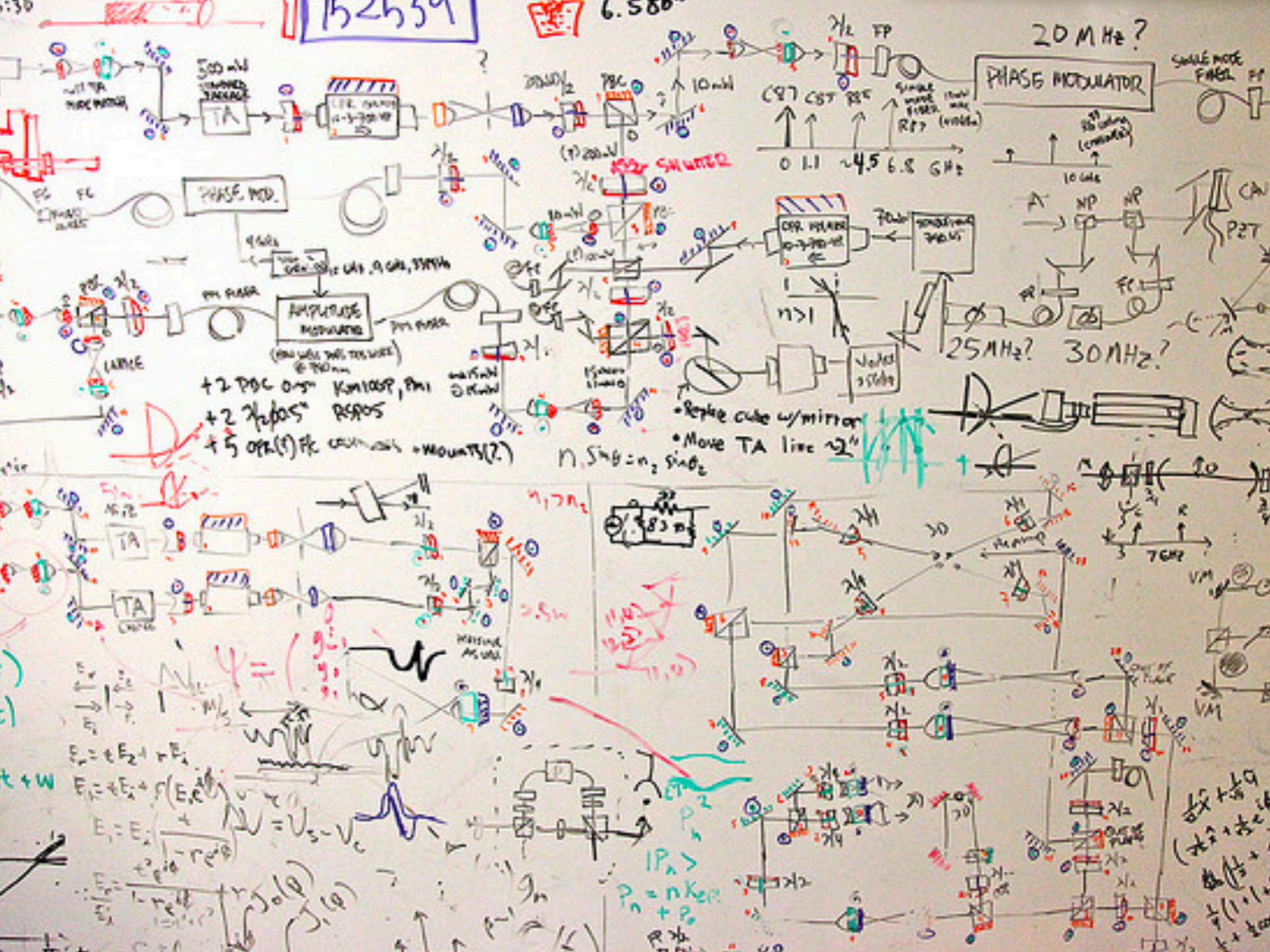


chipie.com



503 Service Temporarily Unavailable

cloudflare



152539

6.580

20 MHz?

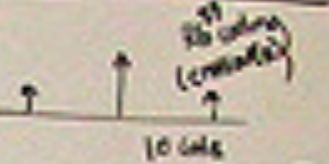
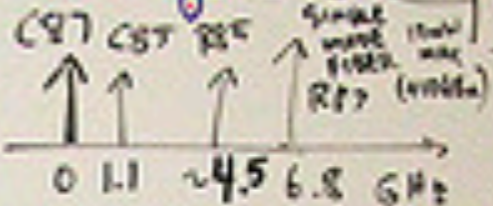
500 mW TA

PHASE MOD.

AMPLITUDE MODULATOR

PHASE MODULATOR

- + 2 PBC 0.5m K=1000, P=1
- + 2 $\lambda/2$ RESOS
- + 5 OFF(T) FC COUPLERS + MOUNTS(?)



25 MHz? 30 MHz?

- Repke cube w/ mirror
- Move TA line $\lambda/2$

$$n_1 \sin \theta_1 = n_2 \sin \theta_2$$

$$E_1 = t E_2 + r E_1$$

$$E_1 = t E_2 + r(E_1 + E_2)$$

$$E_1 = E_2$$

$$t = 1 - r$$

$$t = 1 - r$$

$$t = 1 - r$$

$$t = 1 - r$$

$$t = 1 - r$$

$$t = 1 - r$$

$$t = 1 - r$$

$$IP_n > P_n = n k_e r$$

$$P_n = n k_e r$$

$$P_n = n k_e r$$



Refusal

DATE

SCENE

ACT

04/30

3

1



Refusal

DATE	SCENE	ACT
04/30	3	1



Refusal

DATE

SCENE

ACT

04/30

3

1



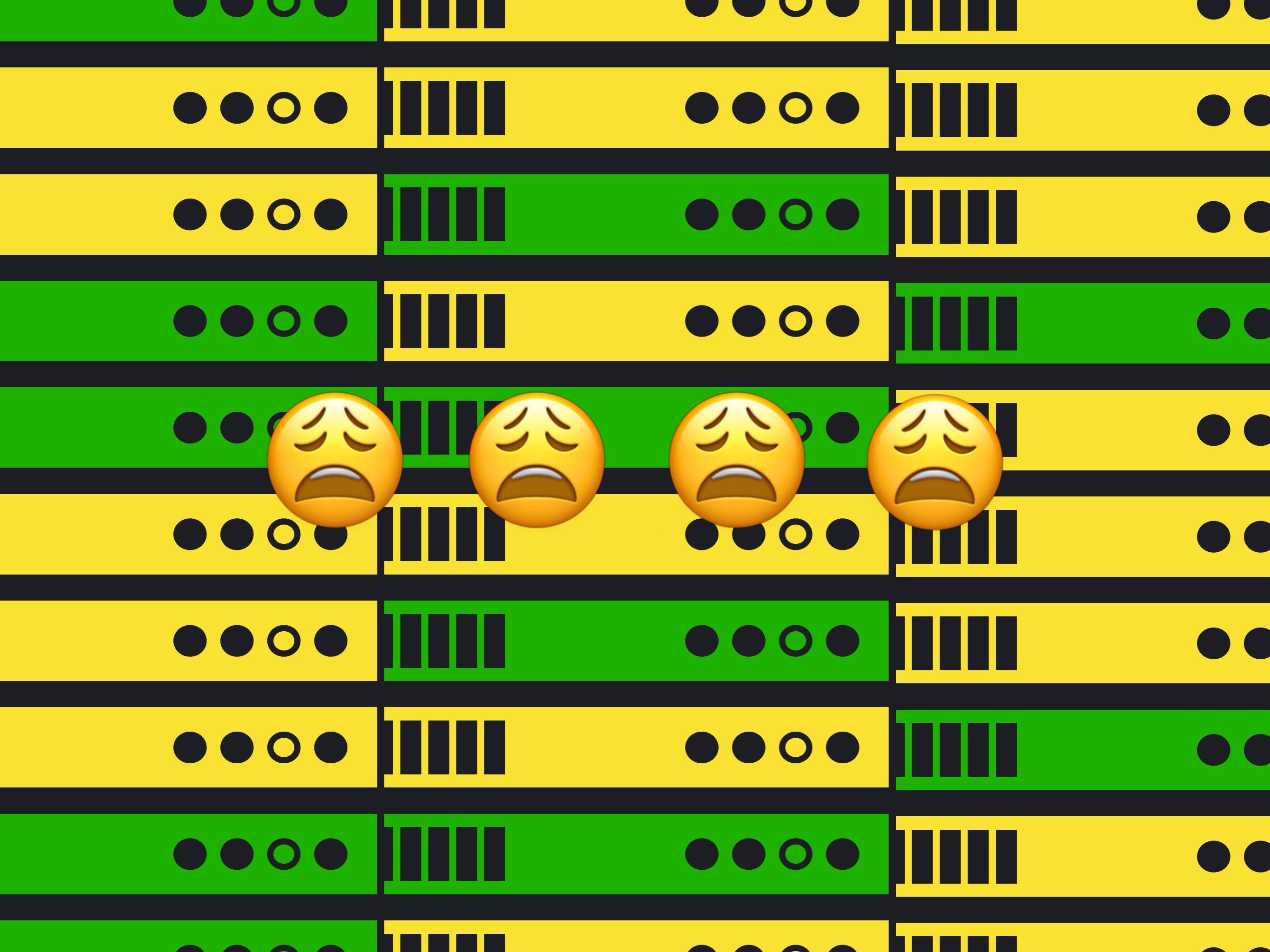
We're gonna need a bigger server rack



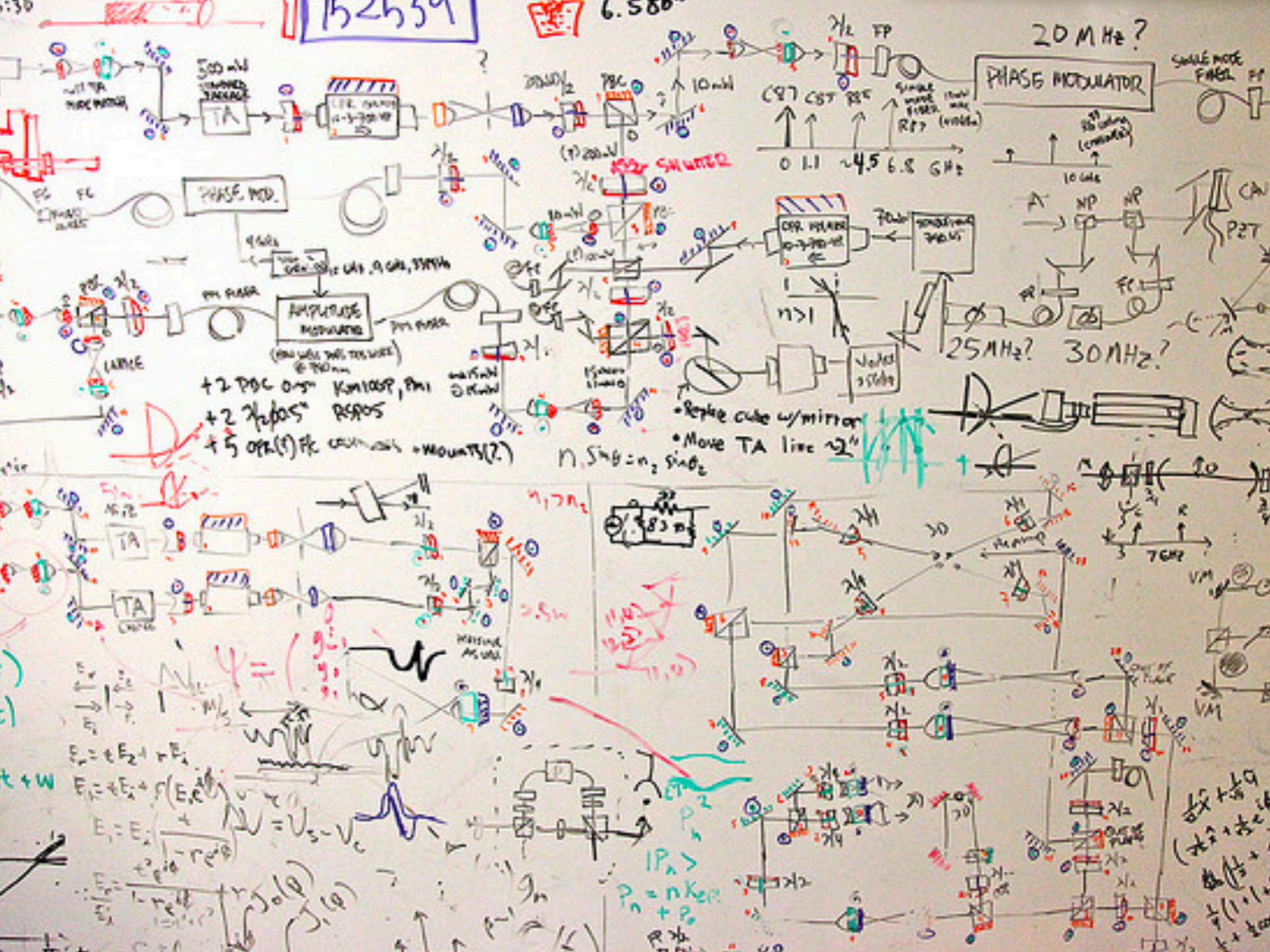












152539

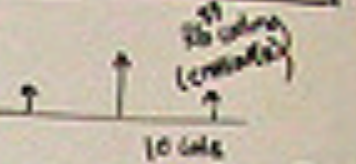
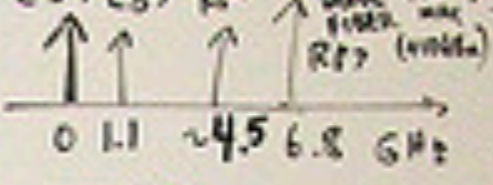
6.580

20 MHz?

500 mW THERMALLY STABILIZED TA

PHASE MODULATOR

SCALE MODE FIBER



PHASE MOD.

AMPLITUDE MODULATOR

25 MHz? 30 MHz?

- + 2 PDC 0.5" KAM100P, Pm1
- + 2 $\lambda/2$ POS RESPOS
- + 5 OPT. Fk COUPLERS + MOUNTS(?)

- Repke cube w/ mirror
- Move TA line 2"

$$n_1 \sin \theta_1 = n_2 \sin \theta_2$$

TA

$$E_1 = tE_2 + rE_1$$

$$E_1 = tE_2 + r(E_1 + E_2)$$

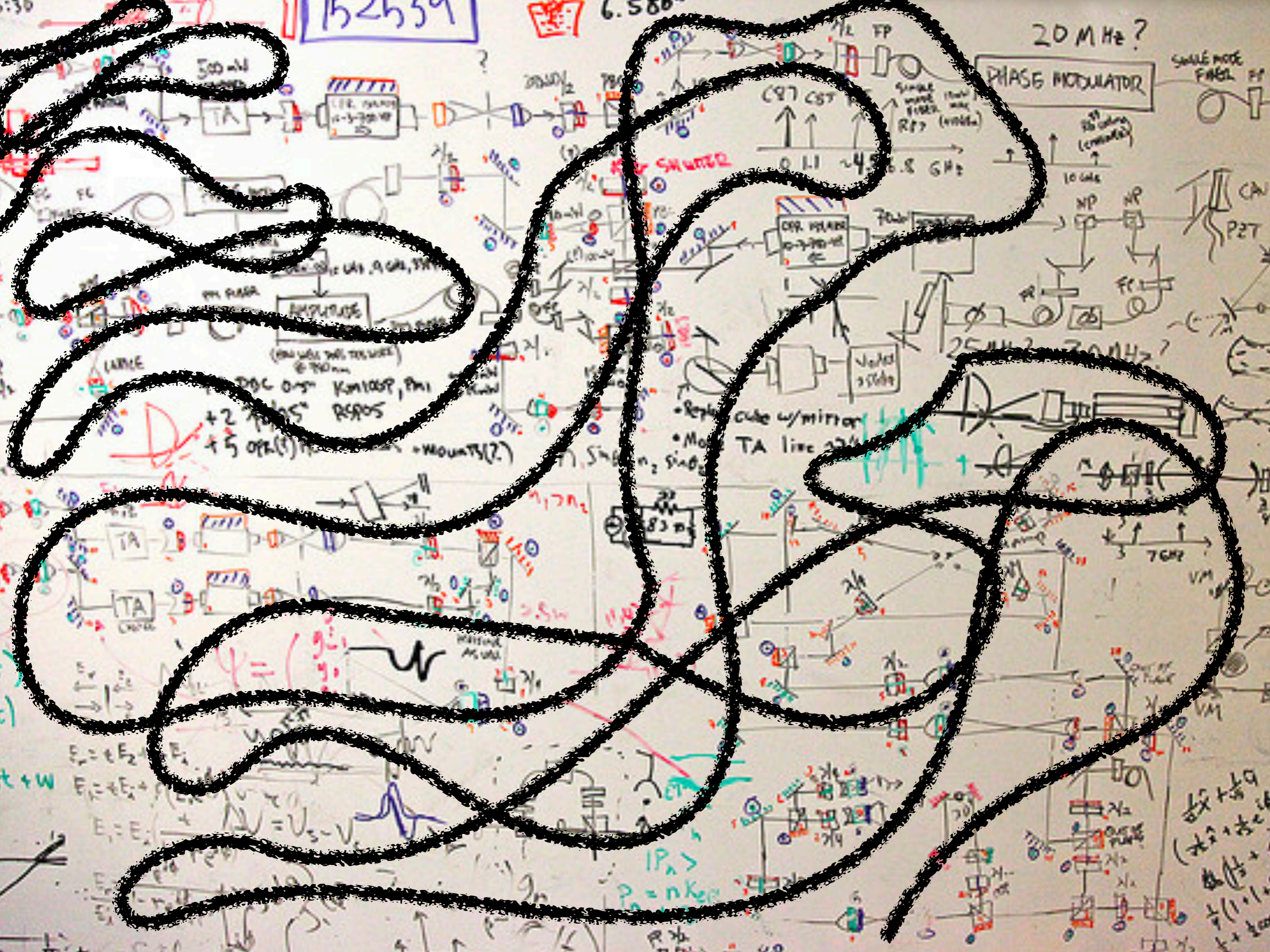
$$E_1 = E_2 + rE_1$$

83

$$IP_n > P_n = n k_e r$$

$$E_1 + E_2 = E_3$$

$$E_1 + E_2 = E_3$$



152539

6.588

20 MHz?

PHASE MODULATOR

SCALE NOTE
FIBER FP

500 mW

TA

1.1E. 10A00
1-3-70-10
5

CS7 CS7
0.11 ~ 4.5 ~ 6.8 GHz

SINGLE MODE FIBER
RFP (VARIABLE)

10 uW

NP

NP

AMP

AMPLITUDE

(10 uW MAX TEST USE)
@ 70 nm

DIG 0-5m KHz/100P, PH1

+2 700 155 RESPOS

+5 OFF(S)

+MOUNTS(?)

• Replic cube w/mirror

• Mod TA line 270

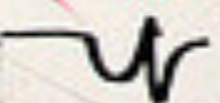
25 MHz? 20 MHz?

700P

700P

VM

$$Y = \frac{1}{1 + j\omega RC}$$



$$E_1 = \epsilon E_2$$

$$E_1 = \epsilon E_2 + \dots$$

$$E_1 = E_2$$

$$V = V_3 - V_4$$

$$IP_n >$$

$$P = n k_B T$$

$$\frac{1}{2} \epsilon \omega^2 + \frac{1}{2} \epsilon \omega^2$$

$$\frac{1}{2} \epsilon \omega^2 + \frac{1}{2} \epsilon \omega^2$$

$$\frac{1}{2} \epsilon \omega^2 + \frac{1}{2} \epsilon \omega^2$$

$$\frac{1}{2} \epsilon \omega^2 + \frac{1}{2} \epsilon \omega^2$$

$$\frac{1}{2} \epsilon \omega^2 + \frac{1}{2} \epsilon \omega^2$$



Meeting the Mentor

DATE SCENE ACT

04/30 4 1



Meeting the Mentor

DATE

SCENE

ACT

04/30

4

1



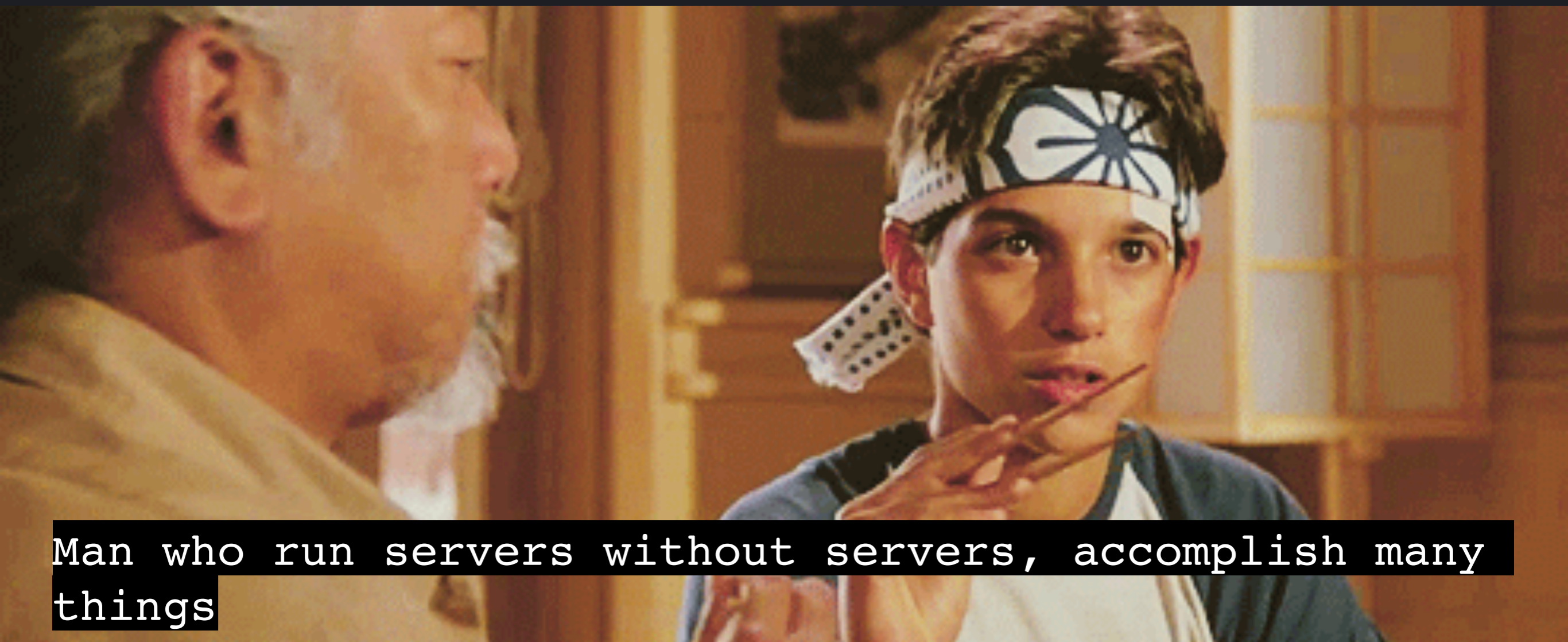
Meeting the Mentor

DATE SCENE ACT

04/30 4 1



Swami Scale-a-lot



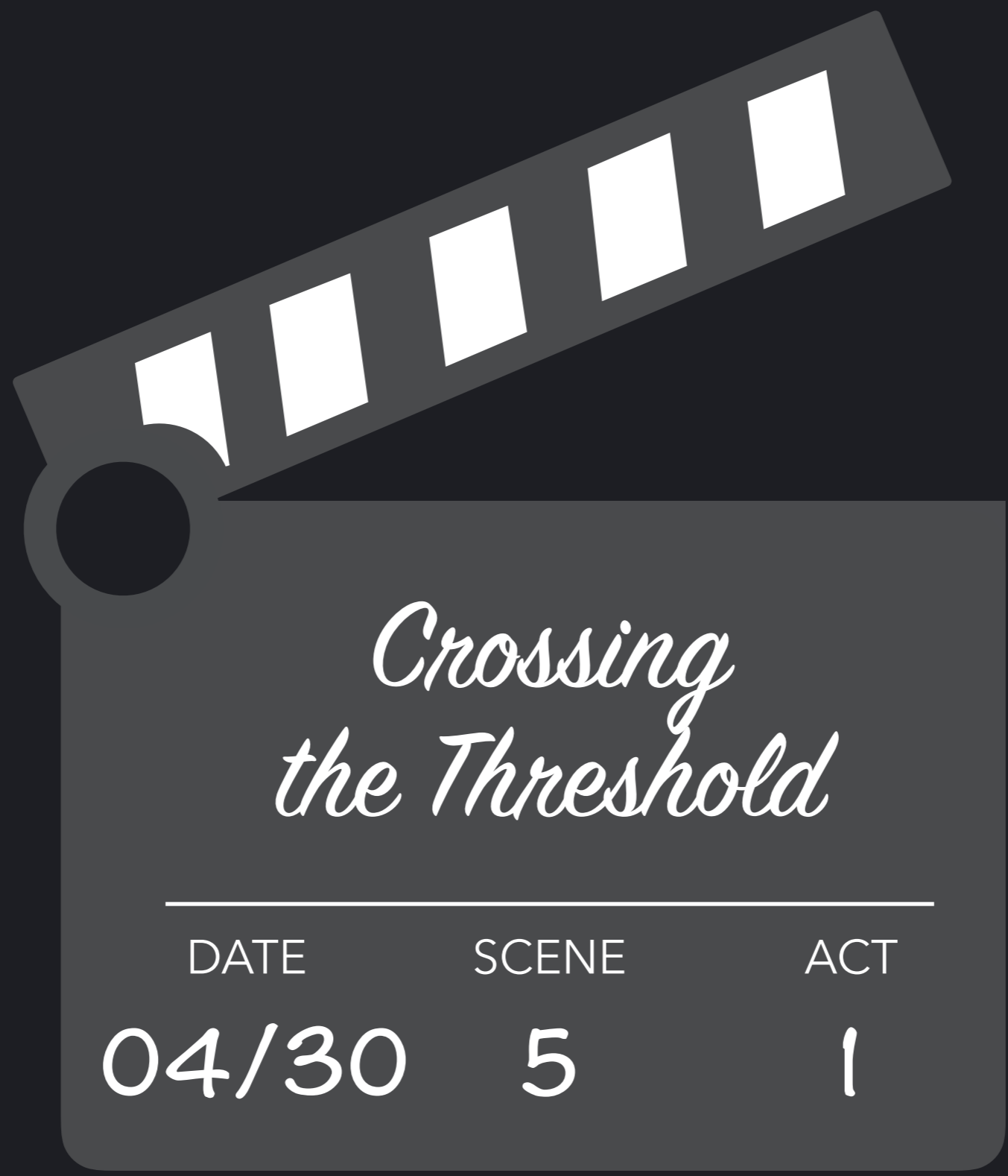
Man who run servers without servers, accomplish many things



*Crossing
the Threshold*

DATE SCENE ACT

04/30 5 1



*Crossing
the Threshold*

DATE	SCENE	ACT
04/30	5	1



*Crossing
the Threshold*

DATE SCENE ACT

04/30 5 1







Act 2

A silhouette of a person walking away from the viewer on a path that recedes into the distance. The path is flanked by a brick wall on the left and a textured wall on the right. The scene is dimly lit, with a bright light source from the left creating a strong shadow and highlighting the path.

Initiation



Tests, Allies, Enemies

DATE

SCENE

ACT

04/30

1

2



Tests, Allies, Enemies

DATE

SCENE

ACT

04/30

1

2



Tests, Allies, Enemies

DATE

SCENE

ACT

04/30

1

2

**THERE
ARE
STILL
SERVERS
IN
SERVERLESS**

(_ _ /) ||

(• 人 •) ||

/ づ

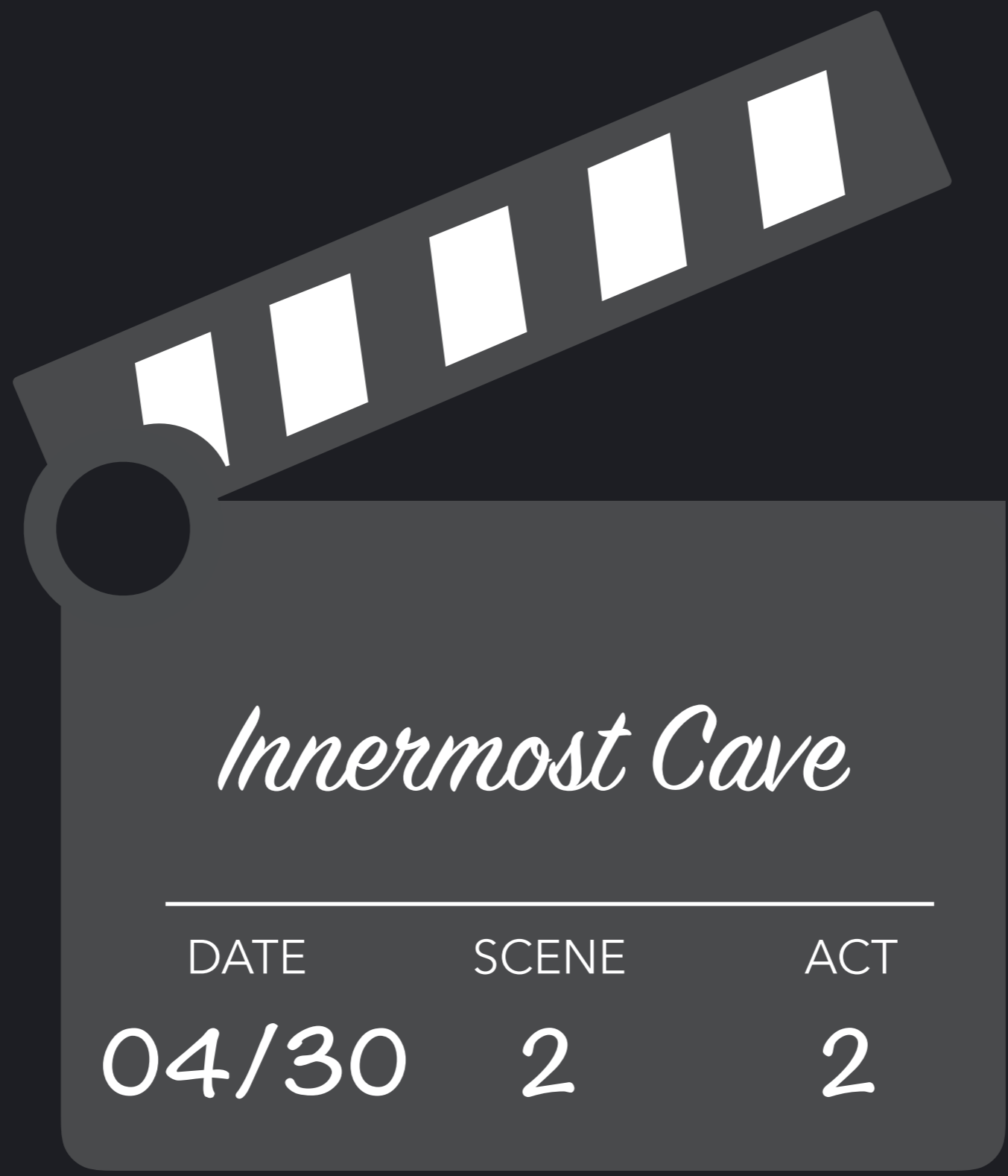


APPROVED



Innermost Cave

DATE	SCENE	ACT
04/30	2	2



Innermost Cave

DATE	SCENE	ACT
04/30	2	2



Innermost Cave

DATE	SCENE	ACT
04/30	2	2







Lou Malnati's Pizzeria



5411 Reviews on Yelp

My Rating:



[Directions](#)

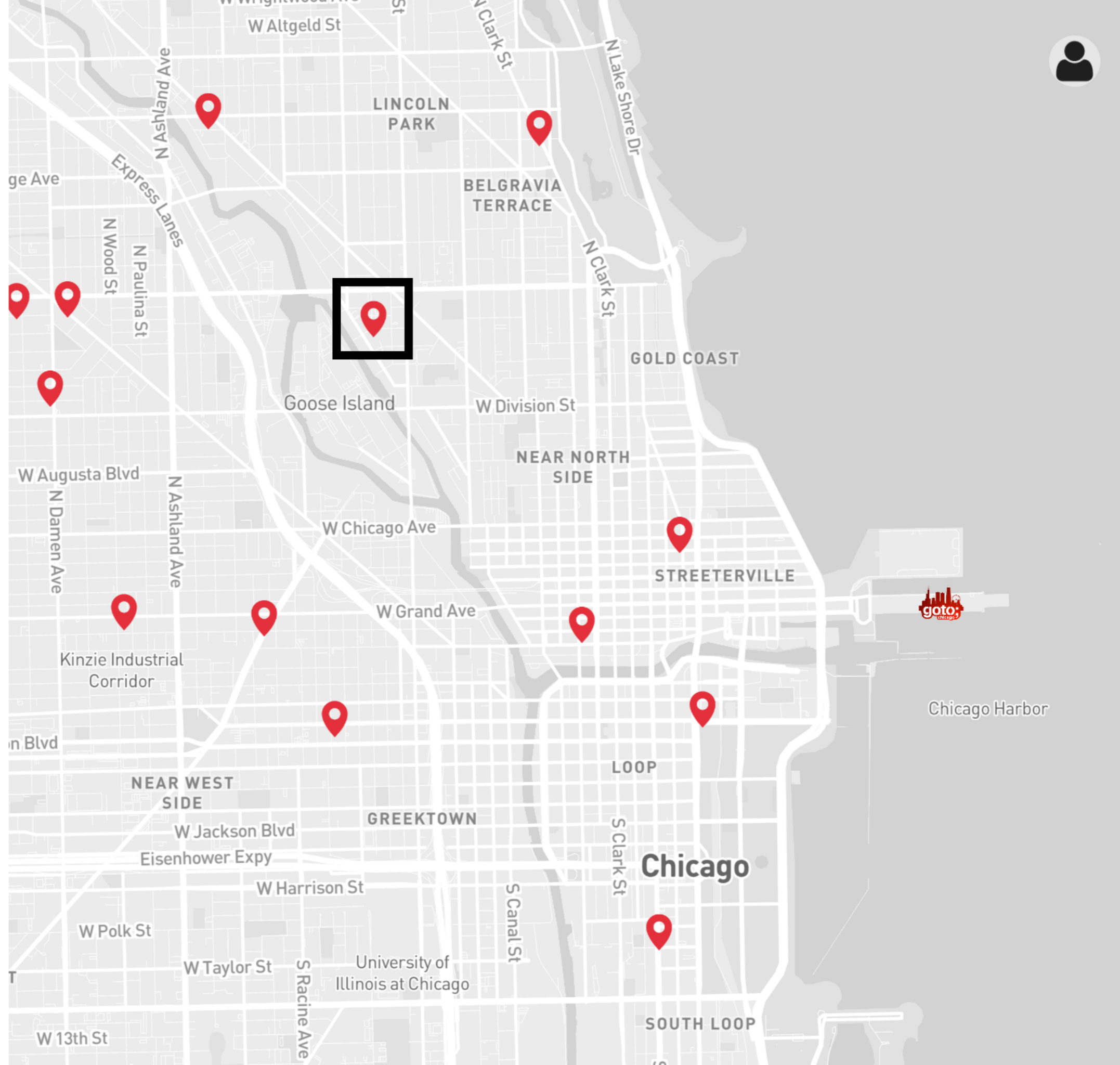


Pequod's Pizzeria



5324 Reviews on Yelp

My Rating:

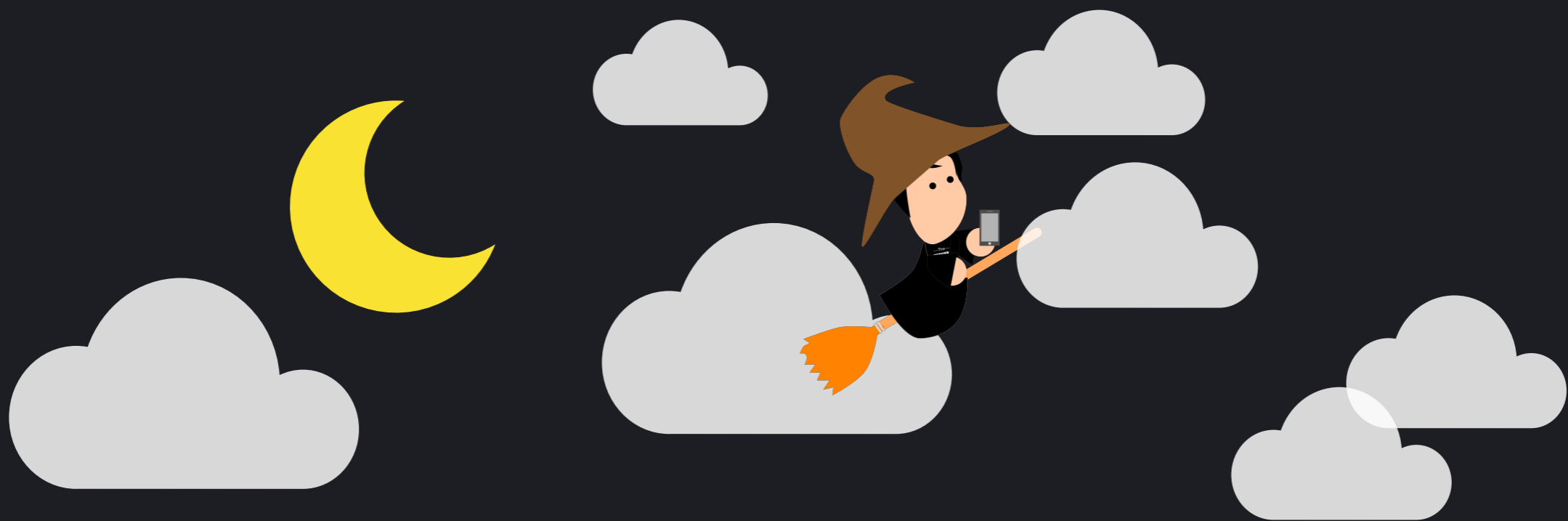



```
1  const express = require('express'),
2     yelp = require('yelp-fusion'),
3     port = 8000,
4     app = express();
5
6  app.use((req, res, next) => {
7    res.header("Access-Control-Allow-Origin", "*");
8    res.header("Access-Control-Allow-Headers", "Origin, X-Requested
9    -with, Content-Type, Accept");
10   next();
11 })
12
13 app.get('/yelpit/:term/:location', (req, res) => {
14   const clientId = CLIENT_ID;
15   const clientSecret = CLIENT_SECRET;
16
17   yelp.accessToken(clientId, clientSecret)
18     .then((response) => {
19     const client = yelp.client(response.jsonBody.access_token);
20     const searchRequest = {
21       term: req.params.term,
22       limit: 50,
23       location: req.params.location
24     }
25     client.search(searchRequest).then(response => {
26     const firstResult = response.jsonBody.businesses;
27     res.send(firstResult)
28   })
29 })
```

```
1  const express = require('express'),
2     yelp = require('yelp-fusion'),
3     port = 8000,
4     app = express();
5
6  app.use((req, res, next) => {
7    res.header("Access-Control-Allow-Origin", "*");
8    res.header("Access-Control-Allow-Headers", "Origin, X-Requested
9    -with, Content-Type, Accept");
10   next();
11 })
12
13 app.get('/yelpit/:term/:location', (req, res) => {
14   const clientId = CLIENT_ID;
15   const clientSecret = CLIENT_SECRET;
16
17   yelp.accessToken(clientId, clientSecret)
18     .then((response) => {
19     const client = yelp.client(response.jsonBody.access_token);
20     const searchRequest = {
21       term: req.params.term,
22       limit: 50,
23       location: req.params.location
24     }
25     client.search(searchRequest).then(response => {
26       const firstResult = response.jsonBody.businesses;
27       res.send(firstResult)
28     })
29   })
30 }
```

```
10  })
11
12
13  app.get('/yelpit/:term/:location', (req, res) => {
14    const clientId = CLIENT_ID;
15    const clientSecret = CLIENT_SECRET;
16
17    yelp.accessToken(clientId, clientSecret)
18      .then((response) => {
19        const client = yelp.client(response.jsonBody.access_token);
20        const searchRequest = {
21          term: req.params.term,
22          limit: 50,
23          location: req.params.location
24        }
25        client.search(searchRequest).then(response => {
26          const firstResult = response.jsonBody.businesses;
27          res.send(firstResult)
28        })
29      })
30  })
31
32  app.listen(port, 'localhost', function(err) {
33    if(err) {console.log(err)}
34    console.info(`Listening on port ${port}`)
35  });
36
```

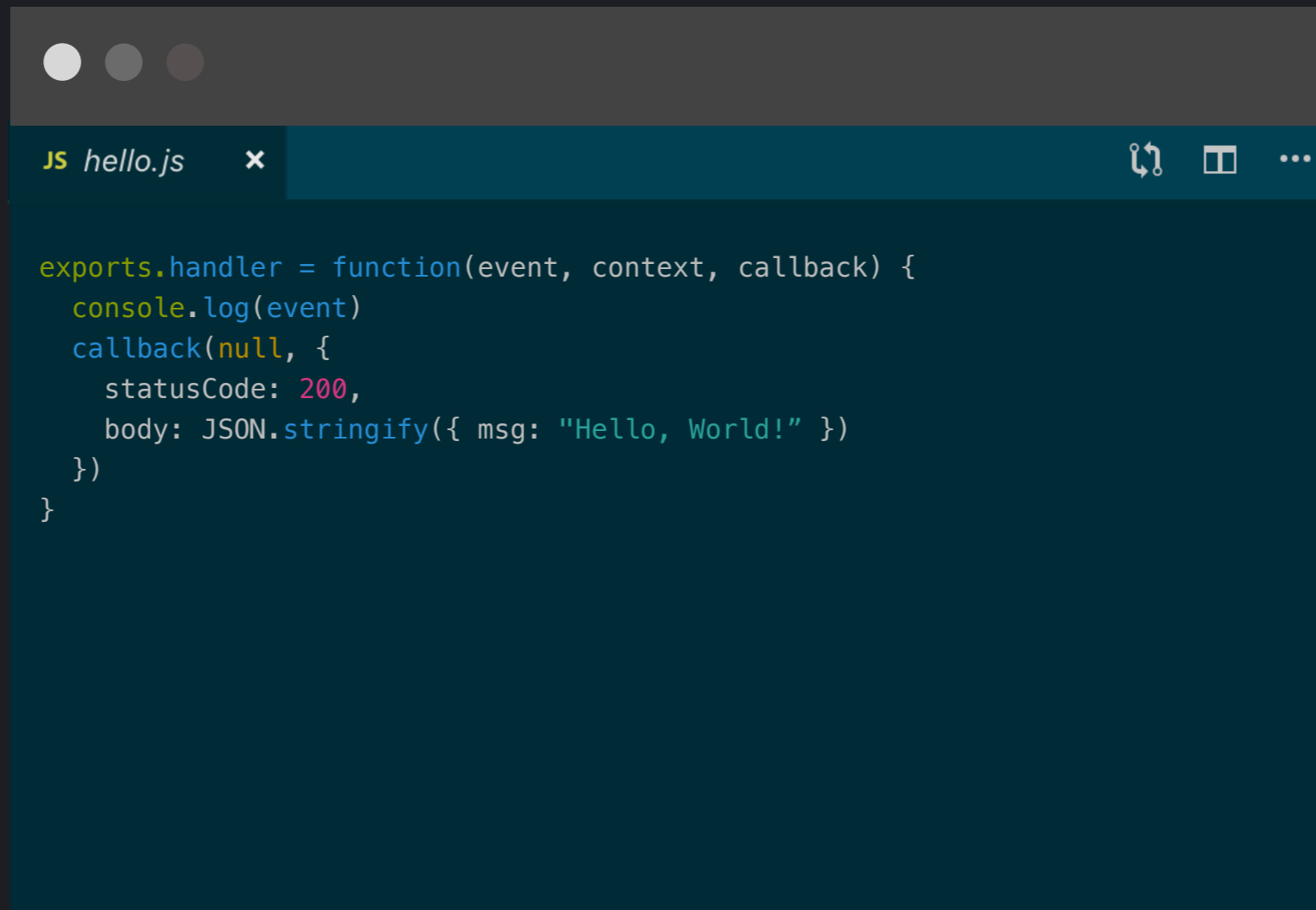
```
1  module.exports = {
2    devServer: {
3      proxy: {
4        '^/yelpit/*': {
5          target: 'http://localhost:8000/yelpit',
6          secure: false
7        }
8      }
9    }
10  }
11
```

The Chilling Adventures of Functions as a Service







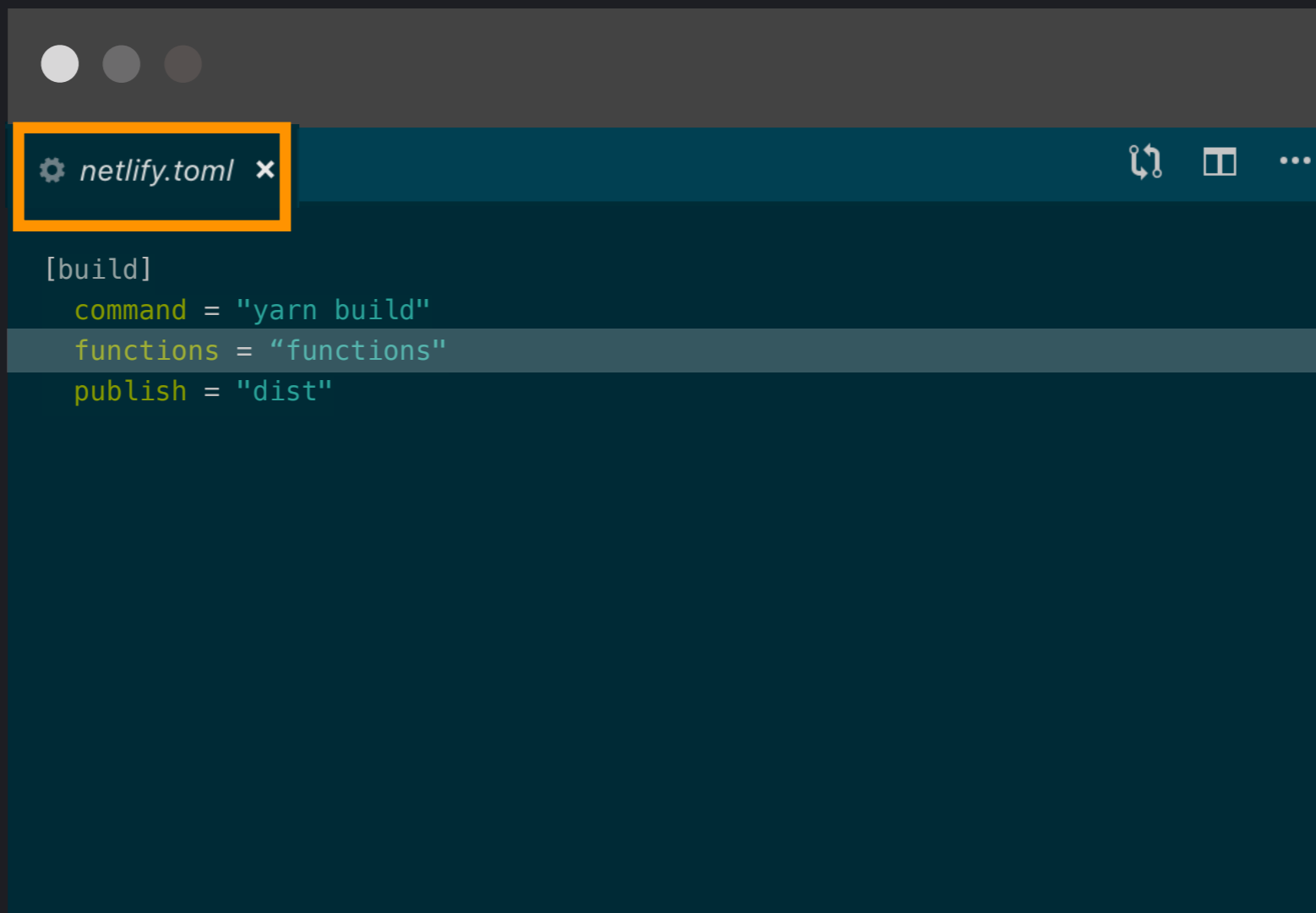
The image shows a code editor window with a dark teal background. The window title bar at the top contains three window control buttons (red, yellow, green) on the left and three icons (refresh, maximize, and menu) on the right. The tab title is 'JS hello.js' with a close button 'x'. The code is as follows:

```
exports.handler = function(event, context, callback) {  
  console.log(event)  
  callback(null, {  
    statusCode: 200,  
    body: JSON.stringify({ msg: "Hello, World!" })  
  })  
}
```

functions/hello.js

```
yarn add netlify-cli
```

```
netlify functions:create
```

```
[build]
  command = "yarn build"
  functions = "functions"
  publish = "dist"
```

netlify.toml



Functions

Functions Free

5 Lambda functions actively running in production.

[Learn more about Functions in the docs](#) ↗

⚙ Settings and usage

🔍 Search by branch or Deploy Preview

handle-signup.js

Created on Mar 14 (19 days ago)



hello.js

Created on Mar 14 (19 days ago)



identity-login.js

Created on Mar 14 (19 days ago)



lock-site.js

Created on Mar 14 (19 days ago)



super-special-function.js



`/.netlify/functions/{function_name}`

`/.netlify/functions/hello`

`/.netlify/functions/yelp-it`

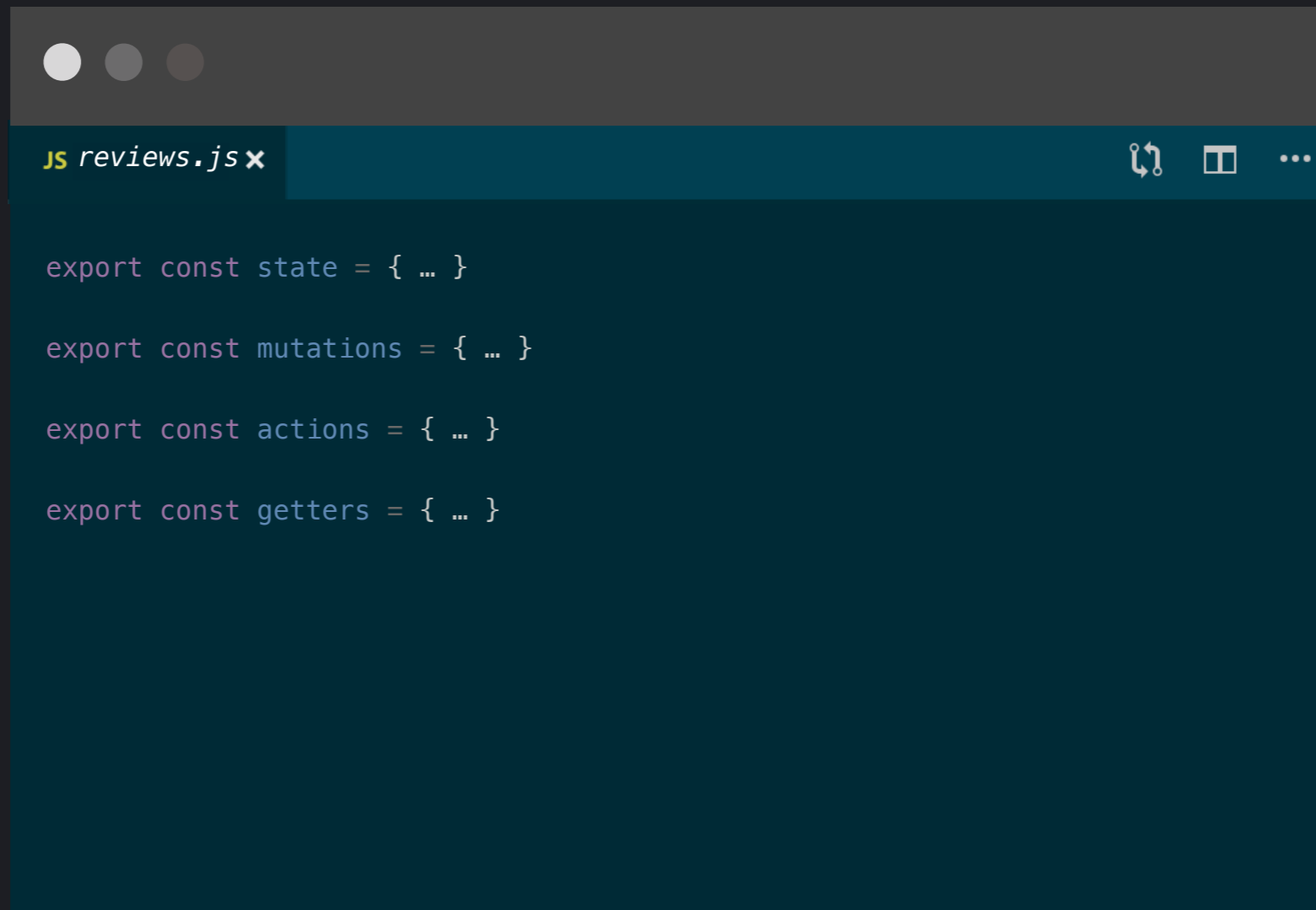

```
1  const yelp = require("yelp-fusion");
2
3  exports.handler = function(event, context, callback) {
4    const apiKey = "API_KEY";
5
6    const client = yelp.client(apiKey);
7
8    const { term, location } = event.queryStringParameters;
9
10   const searchRequest = {
11     term: term,
12     location: location
13   };
14
15   var fetchFromYelp = async function() {
16     try {
17       let res = await client.search(searchRequest);
18       callback(null, {
19         statusCode: 200,
20         body: JSON.stringify({
21           results: res
22         })
23       });
24     } catch (err) {
25       callback(null, {
26         statusCode: 200,
27         body: JSON.stringify({
28           err: err
29         })
30     }
```

```
10  const searchRequest = {
11    term: term,
12    location: location
13  };
14
15  var fetchFromYelp = async function() {
16    try {
17      let res = await client.search(searchRequest);
18      callback(null, {
19        statusCode: 200,
20        body: JSON.stringify({
21          results: res
22        })
23      });
24    } catch (err) {
25      callback(null, {
26        statusCode: 200,
27        body: JSON.stringify({
28          err: err
29        })
30      });
31    }
32  };
33
34  fetchFromYelp();
35 };
36
```

```
1 <template>
2   <div class="home">
3     <div id="sidebar" v-if="dataLoaded">
4       <Slider
5         :yelpData="yelpData.features"
6       />
7     </div>
8   </div>
9 </template>
10
11 <script>
12 export default {
13   name: "home",
14   mounted() {
15     axios
16       .get("/.netlify/functions/yelp-it", {
17         params: {
18           location: "chicago,il",
19           term: "pizza"
20         }
21       })
22       .then(async res => {
23         res = JSON.parse(res.data.results.body);
24         const results = this.geojsonify(res.businesses);
25         this.dataLoaded = true;
26         this.yelpData = results;
27       },
28     },
29
```

```
1 <template>
2   <div class="home">
3     <div id="sidebar" v-if="dataLoaded">
4       <Slider
5         :yelpData="yelpData.features"
6       />
7     </div>
8   </div>
9 </template>
10
11 <script>
12 export default {
13   name: "home",
14   mounted() {
15     axios
16       .get("/.netlify/functions/yelp-it", {
17         params: {
18           location: "chicago,il",
19           term: "pizza"
20         }
21       })
22     .then(async res => {
23       res = JSON.parse(res.data.results.body);
24       const results = this.geojsonify(res.businesses);
25       this.dataLoaded = true;
26       this.yelpData = results;
27     },
28     },
29
```

```
11 <script>
12 export default {
13   name: "home",
14   mounted() {
15     axios
16       .get("/.netlify/functions/yelp-it", {
17         params: {
18           location: "chicago,il",
19           term: "pizza"
20         }
21       })
22     .then(async res => {
23       res = JSON.parse(res.data.results.body);
24       const results = this.geojsonify(res.businesses);
25       this.dataLoaded = true;
26       this.yelpData = results;
27     },
28     methods: {
29       geojsonify(response) {
30         let features = [];
31         response.map(item => {
32           features.push({
33             type: "Feature",
34             geometry: {
35               type: "Point",
36               coordinates: [item.coordinates.longitude,
37                 item.coordinates.latitude]
38             },
39             properties: {
```

The image shows a code editor window with a dark teal background. At the top, there is a title bar with three window control buttons (minimize, maximize, close) on the left and three utility icons (refresh, split, and menu) on the right. The title bar text reads "JS reviews.js x". The main area of the editor contains the following JavaScript code:

```
export const state = { ... }  
  
export const mutations = { ... }  
  
export const actions = { ... }  
  
export const getters = { ... }
```

src/state/reviews.js

```
1 export const state = {
2   yelpData: null
3 };
4
5 export const mutations = {
6   SET_YELP_REVIEWS(state, value) {
7     state.yelpData = value
8   },
9 }
10 export const actions = {
11   getYelp({ commit }, value) {
12     var geojsonify = function(data) { ... }
13
14     axios
15       .get("/.netlify/functions/yelp-it", {
16         params: {
17           location: "chicago,il",
18           term: "pizza"
19         }
20       })
21       .then(async res => {
22         res = JSON.parse(res.data.results.body);
23         const results = geojsonify(res.businesses);
24         commit("SET_YELP_REVIEWS", res.docs)
25       })
26   }
27 }
28
29
30
```



```
1 export const state = {
2   yelpData: null
3 };
4
5 export const mutations = {
6   SET_YELP_REVIEWS(state, value) {
7     state.yelpData = value
8   },
9 }
10 export const actions = {
11   getYelp({ commit }, value) {
12     var geojsonify = function(data) { ... }
13
14     axios
15       .get("/.netlify/functions/yelp-it", {
16         params: {
17           location: "chicago,il",
18           term: "pizza"
19         }
20       })
21       .then(async res => {
22         res = JSON.parse(res.data.results.body);
23         const results = geojsonify(res.businesses);
24         commit("SET_YELP_REVIEWS", res.docs)
25       })
26   }
27 }
28
29
30
```

```
1 export const state = {
2   yelpData: null
3 };
4
5 export const mutations = {
6   SET_YELP_REVIEWS(state, value) {
7     state.yelpData = value
8   },
9 }
10 export const actions = {
11   getYelp({ commit }, value) {
12     var geojsonify = function(data) { ... }
13
14     axios
15       .get("/.netlify/functions/yelp-it", {
16         params: {
17           location: "chicago,il",
18           term: "pizza"
19         }
20       })
21       .then(async res => {
22         res = JSON.parse(res.data.results.body);
23         const results = geojsonify(res.businesses);
24         commit("SET_YELP_REVIEWS", res.docs)
25       })
26   }
27 }
28
29
30
```



```
1 export const state = {
2   yelpData: null
3 };
4
5 export const mutations = {
6   SET_YELP_REVIEWS(state, value) {
7     state.yelpData = value
8   },
9 }
10 export const actions = {
11   getYelp({ commit }, value) {
12     var geojsonify = function(data) { ... }
13
14     axios
15       .get("/.netlify/functions/yelp-it", {
16         params: {
17           location: "chicago,il",
18           term: "pizza"
19         }
20       })
21       .then(async res => {
22         res = JSON.parse(res.data.results.body);
23         const results = geojsonify(res.businesses);
24         commit("SET_YELP_REVIEWS", res.docs)
25       })
26   }
27 }
28
29
30
```

```
1 <template>
2   <div class="home">
3     <div id="sidebar" v-if="dataLoaded">
4       <Slider
5         :yelpData="yelpData.features"
6       />
7     </div>
8   </div>
9 </template>
10
11 <script>
12 import { mapState, mapActions } from "vuex";
13
14 export default {
15   name: "home",
16   computed: {
17     ...mapState("yelp", ["yelpData"])
18   },
19   methods: {
20     ...mapActions("reviews", ["getYelp"]),
21   },
22   mounted() {
23     this.getYelp()
24   }
25 }
26 </script>
27
28
29
30
```

```
1 <template>
2   <div class="home">
3     <div id="sidebar" v-if="dataLoaded">
4       <Slider
5         :yelpData="yelpData.features"
6       />
7     </div>
8   </div>
9 </template>
```

```
10
11 <script>
12 import { mapState, mapActions } from "vuex";
13
14 export default {
15   name: "home",
16   computed: {
17     ...mapState("yelp", ["yelpData"])
18   },
19   methods: {
20     ...mapActions("reviews", ["getYelp"]),
21   },
22   mounted() {
23     this.getYelp()
24   }
25 }
26 </script>
```

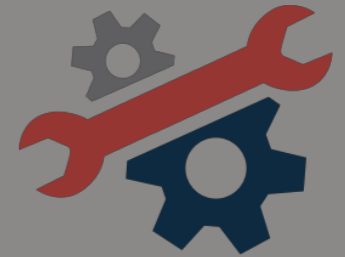
27

28

29

30

```
1 <template>
2   <div class="home">
3     <div id="sidebar" v-if="dataLoaded">
4       <Slider
5         :yelpData="yelpData.features"
6       />
7     </div>
8   </div>
9 </template>
10
11 <script>
12 import { mapState, mapActions } from "vuex";
13
14 export default {
15   name: "home",
16   computed: {
17     ...mapState("yelp", ["yelpData"])
18   },
19   methods: {
20     ...mapActions("reviews", ["getYelp"]),
21   },
22   mounted() {
23     this.getYelp()
24   }
25 }
26 </script>
27
28
29
30
```





Lou Malnati's Pizzeria



5411 Reviews on Yelp

My Rating:



[Directions](#)

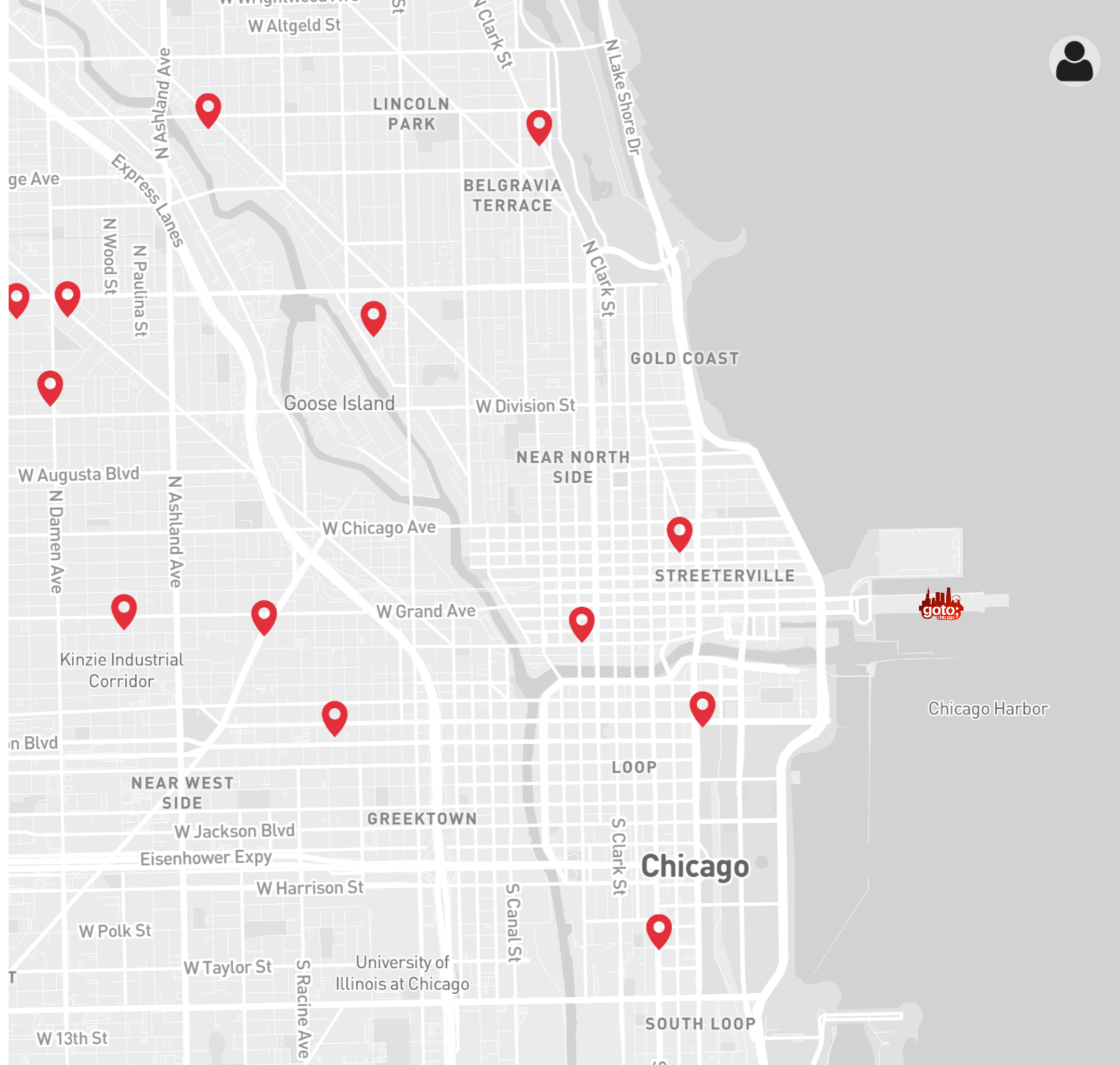


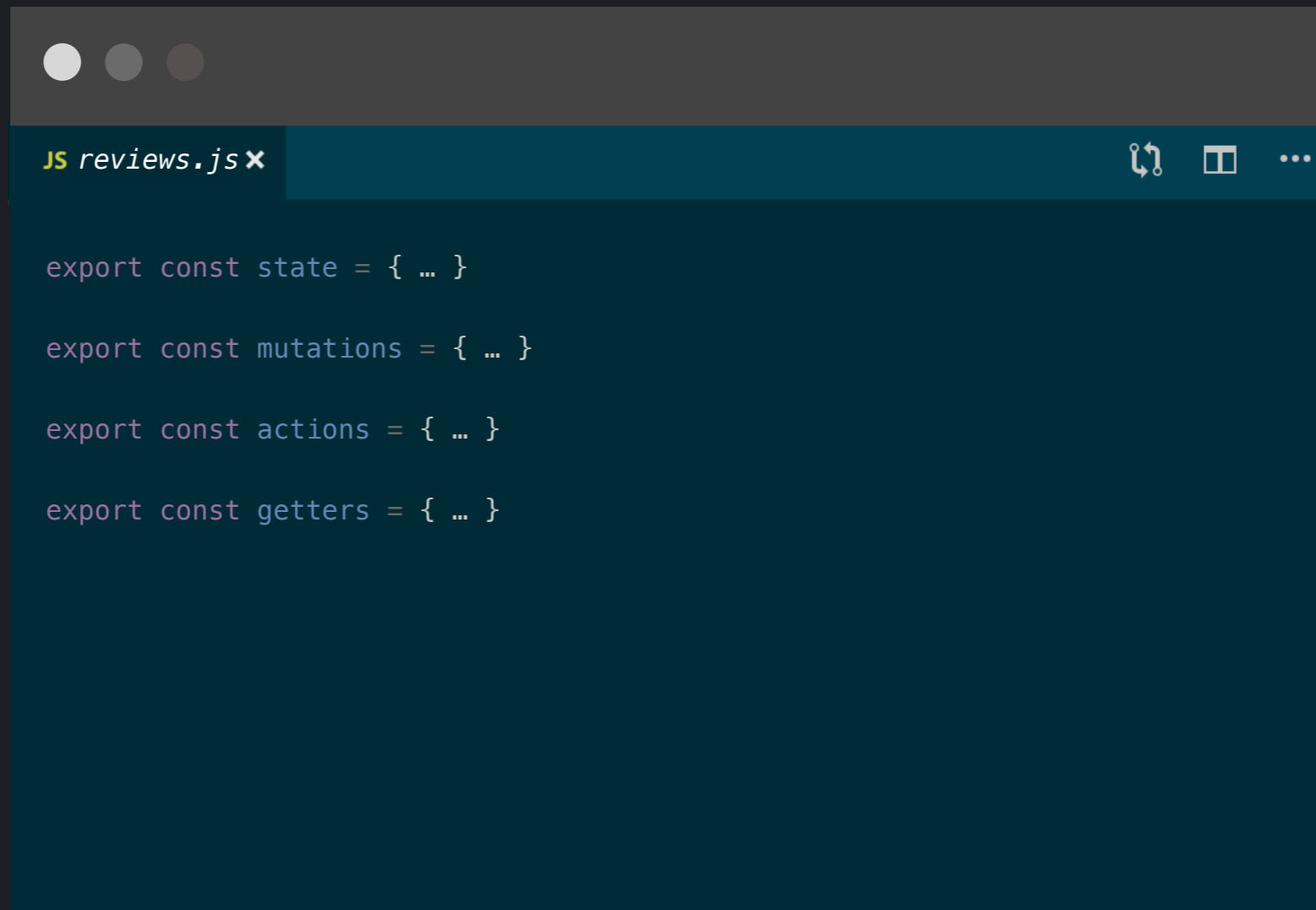
Pequod's Pizzeria



5324 Reviews on Yelp

My Rating:





```
JS reviews.js x [refresh] [grid] [menu]  
  
export const state = { ... }  
  
export const mutations = { ... }  
  
export const actions = { ... }  
  
export const getters = { ... }
```

src/state/reviews.js


```
1 firebase.initializeApp(config);
2 const db = firebase.firestore();
```

```
3
4 export const state = {
5   chiPieReviews: {}
6 };
7
```

```
8 export const mutations = {
9   SET_CHIPIE_REVIEWS(state, value) {
10     var jsonify = function() {}
11     var t = {}
12     value.forEach(val => {
13       t[val.id] = val.data()
14     })
15     debugger
16     state.chiPieReviews = t
17   }
18 };
19
```

```
20 export const actions = {
21   createRating({ commit }, value) {
22     return new Promise((resolve, reject) => {
23       var user = db
24         .collection("users")
25         .doc(value.id);
26       user
27         .collection("pizza-places")
28         .doc(value.name)
29         .set({
```

```
74 },
75 setRatings({ dispatch }, value) {
76   return new Promise((resolve, reject) => {
77     var user = db
78       .collection("users")
79       .doc(value.id);
80     var docRef = user.collection("pizza-places").doc(value.name);
81
82     docRef
83       .get()
84       .then(doc => {
85         if (doc.exists) {
86           resolve("oh yeah");
87           dispatch("updateRating", value);
88         } else {
89           dispatch("createRating", value);
90           resolve("empty");
91         }
92       })
93       .catch(err => {
94         reject(err);
95       });
96   });
97 },
98 getRatings({ commit }, value) {
99   return new Promise((resolve, reject) => {
100     var user = db
101       .collection("users")
102       .doc(value);
103     user
```



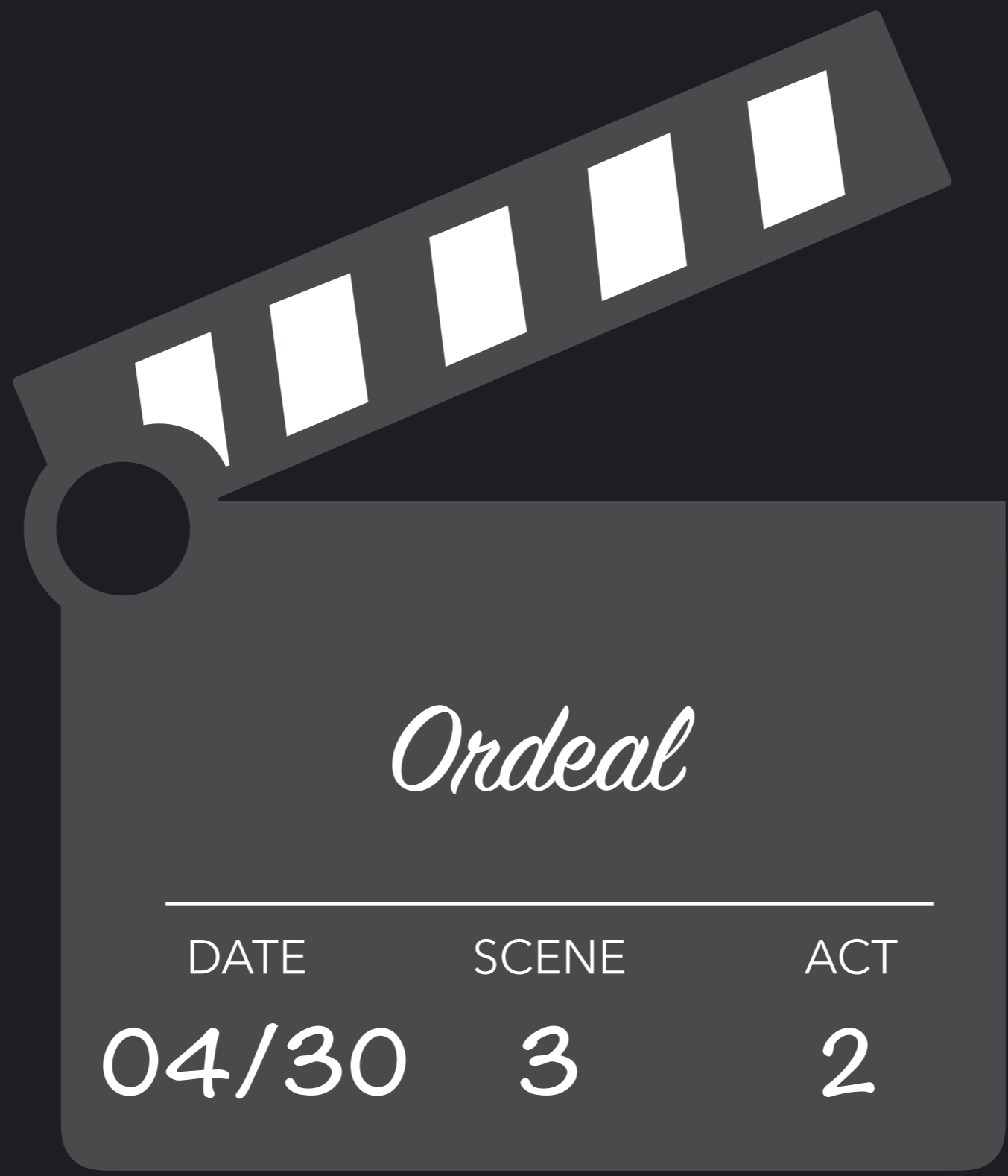
Ordeal

DATE SCENE ACT

04/30

3

2



Ordeal

DATE	SCENE	ACT
04/30	3	2



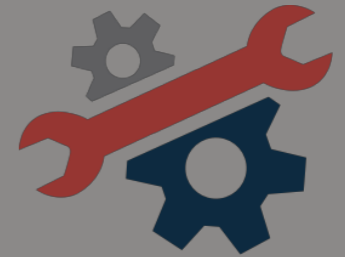
Ordeal

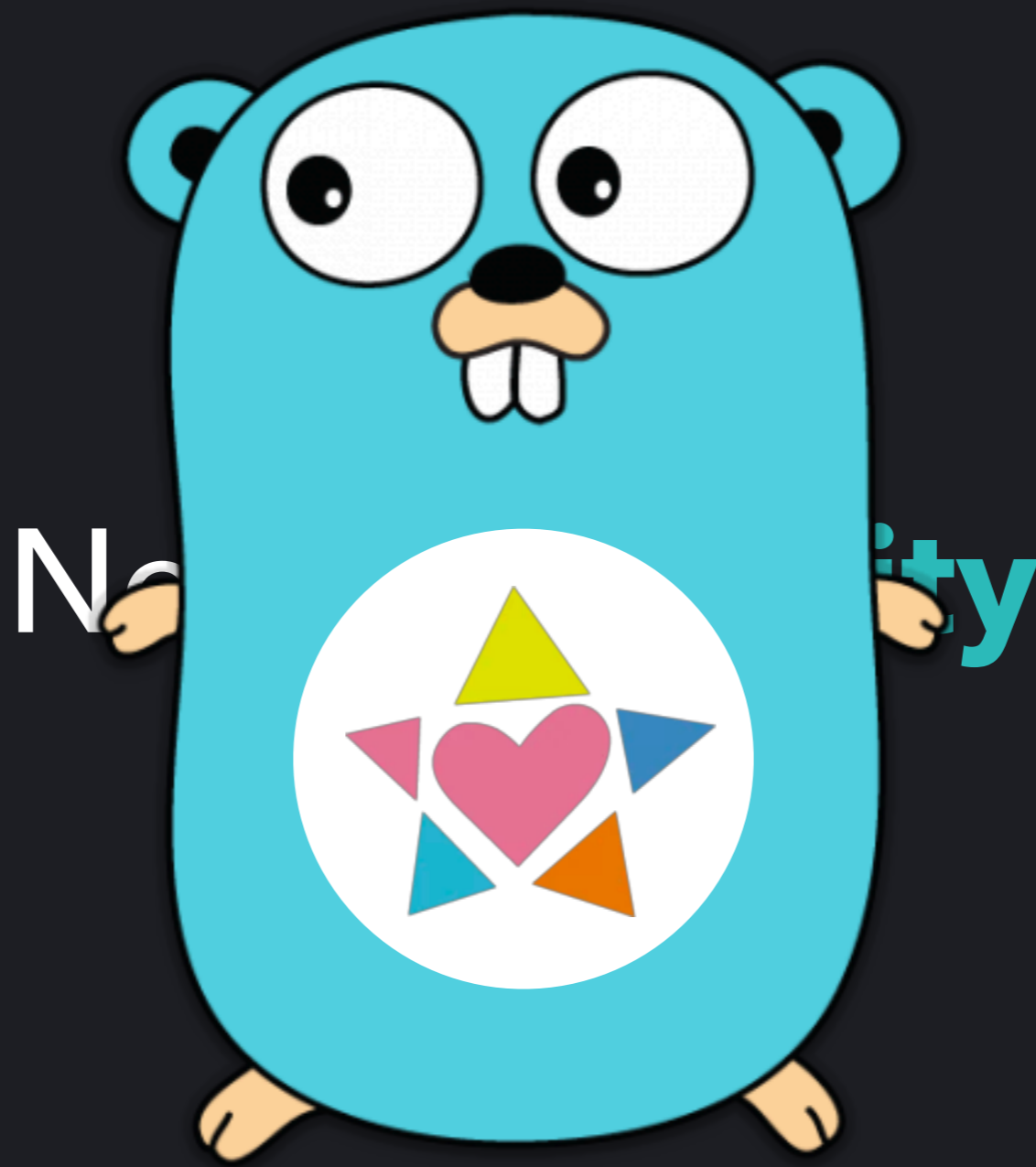
DATE SCENE ACT

04/30

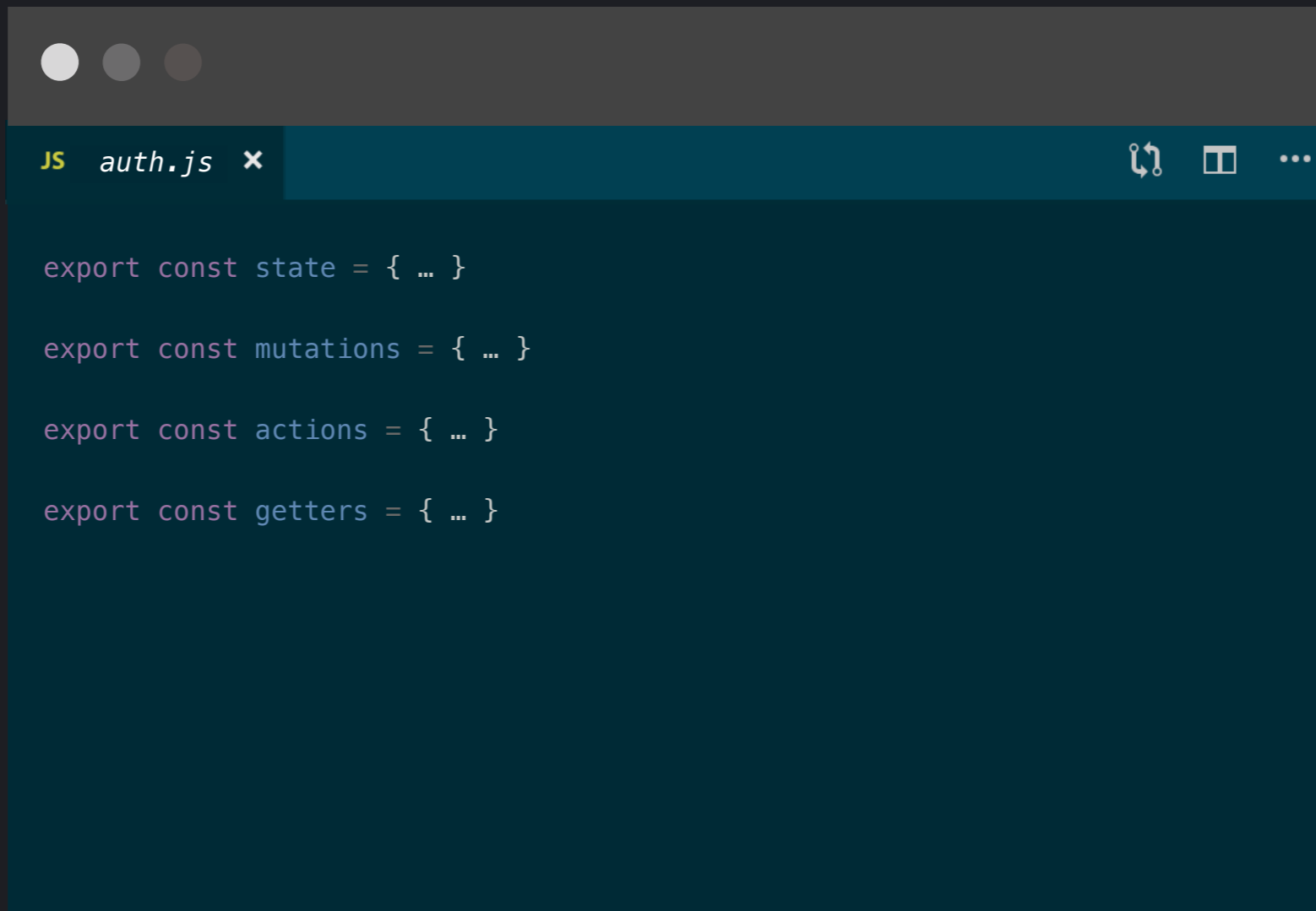
3

2





GoTrue



The image shows a code editor window with a dark teal background. At the top, there are three window control buttons (minimize, maximize, close) on the left and three utility icons (refresh, split, and menu) on the right. The tab bar shows a single tab labeled 'JS auth.js' with a close button. The code content is as follows:

```
export const state = { ... }  
  
export const mutations = { ... }  
  
export const actions = { ... }  
  
export const getters = { ... }
```

src/state/auth.js

```
1 import GoTrue from "gotrue-js";
2 import axios from "axios";
3
4 const auth = new GoTrue({
5   APIUrl: "https://chipie.netlify.com/.netlify/identity",
6   audience: "",
7   setCookie: false
8 });
9
10
11 export const state = {
12   currentUser: getSavedState("auth.currentUser"),
13   loading: false,
14   token: null,
15   notifications: []
16 };
17
18 export const mutations = {
19   SET_CURRENT_USER(state, value) {
20     state.currentUser = value;
21     saveState("auth.currentUser", value);
22   },
23   TOGGLE_LOAD(state) {
24     state.loading = !state.loading;
25   }
26 };
27
28 export const getters = {
29   isLoggedIn(state) {
```

```
1 import GoTrue from "gotrue-js";
2 import axios from "axios";
3
4 const auth = new GoTrue({
5   APIUrl: "https://chipie.netlify.com/.netlify/identity",
6   audience: "",
7   setCookie: false
8 });
9
10
11 export const state = {
12   currentUser: getSavedState("auth.currentUser"),
13   loading: false,
14   token: null,
15   notifications: []
16 };
17
18 export const mutations = {
19   SET_CURRENT_USER(state, value) {
20     state.currentUser = value;
21     saveState("auth.currentUser", value);
22   },
23   TOGGLE_LOAD(state) {
24     state.loading = !state.loading;
25   }
26 };
27
28 export const getters = {
29   isLoggedIn(state) {
```



```
32 export const actions = {
33   init() {
34     localStorage.removeItem("auth.currentUser");
35   },
36   validate({ commit, state }) {
37     if (!state.currentUser) return Promise.resolve(null);
38     const user = auth.currentUser();
39     commit("SET_CURRENT_USER", user);
40     return user;
41   },
42   attemptLogin({ commit, dispatch }, credentials) {
43     return new Promise((resolve, reject) => {
44       dispatch("attemptConfirmation", credentials).then(() => {
45         auth
46           .login(credentials.email, credentials.password)
47           .then(response => {
48             resolve(response);
49             commit("SET_CURRENT_USER", response);
50           })
51           .catch(error => {
52             reject(error.json);
53           });
54       });
55     });
56   },
57
58   attemptConfirmation({ commit, dispatch }, credentials) {
59     return new Promise((resolve, reject) => {
60       if (!credentials.token) {
```

```
15
16 export const mutations = {
17   SET_CURRENT_USER(state, value) {
18     state.currentUser = value;
19     saveState("auth.currentUser", value);
20   },
21   TOGGLE_LOAD(state) {
22     state.loading = !state.loading;
23   }
24 };
25
26 export const getters = {
27   isLoggedIn(state) {
28     return !!state.currentUser;
29   }
30 };
31
32 export const actions = {
33   init() {
34     localStorage.removeItem("auth.currentUser");
35   },
36   validate({ commit, state }) {
37     if (!state.currentUser) return Promise.resolve(null);
38     const user = auth.currentUser();
39     commit("SET_CURRENT_USER", user);
40     return user;
41   },
42   attemptLogin({ commit, dispatch }, credentials) {
43     return new Promise((resolve, reject) => {
44       dispatch("attemptConfirmation", credentials).then(() => {
```

```
1 <template>
2   <div class="login-screen">
3     <div class="account-login">
4       <form @submit.prevent="login()">
5         <label>
6           <span>Email:</span>
7           <input
8             type="text"
9             placeholder="name"
10            v-model="loginCreds.email"
11          />
12        </label>
13        <label>
14          <span>Password:</span>
15          <input
16            type="password"
17            placeholder="password"
18            v-model="loginCreds.password"
19          />
20        </label>
21        <button type="submit" class="account-button">Login</button>
22      </form>
23    </div>
24  </div>
25 </template>
26
27 <script>
28 import { mapActions } from "vuex";
29
```



```
1 <template>
2   <div class="login-screen">
3     <div class="account-login">
4       <form @submit.prevent="login()">
5         <label>
6           <span>Email:</span>
7           <input
8             type="text"
9             placeholder="name"
10            v-model="loginCreds.email"
11          />
12        </label>
13        <label>
14          <span>Password:</span>
15          <input
16            type="password"
17            placeholder="password"
18            v-model="loginCreds.password"
19          />
20        </label>
21        <button type="submit" class="account-button">Login</button>
22      </form>
23    </div>
24  </div>
25 </template>
26
27 <script>
28 import { mapActions } from "vuex";
29
```

```
26
27 <script>
28 import { mapActions } from "vuex";
29
30 export default {
31   name: "LoginAccount",
32   data() {
33     return {
34       isNewUser: true,
35       loginCreds: {
36         email: null,
37         password: null
38       }
39     },
40   },
41   methods: {
42     ...mapActions("auth", ["attemptLogin"]),
43     transferToDashboard() {
44       this.$router.push(this.$route.query.redirect || "/");
45     },
46     login() {
47       let token = decodeURIComponent(window.location.search)
48         .substring(1)
49         .split("confirmation_token=")[1];
50       this.attemptLogin({ token, ...this.loginCreds })
51         .then(res => {
52           this.transferToDashboard();
53         })
54         .catch(err => {
55           console.log(err);
56         });
57     }
58   }
59 }
```

```
40 },
41 methods: {
42   ...mapActions("auth", ["attemptLogin"]),
43   transferToDashboard() {
44     this.$router.push(this.$route.query.redirect || "/");
45   },
46   login() {
47     let token = decodeURIComponent(window.location.search)
48       .substring(1)
49       .split("confirmation_token=")[1];
50     this.attemptLogin({ token, ...this.loginCreds })
51       .then(res => {
52         this.transferToDashboard();
53       })
54       .catch(err => {
55         console.log(err);
56       });
57   },
58 }
59 }
60 </script>
61
62
63
```

```
40 },
41 methods: {
42   ...mapActions("auth", ["attemptLogin"]),
43   transferToDashboard() {
44     this.$router.push(this.$route.query.redirect || "/");
45   },
46   login() {
47     let token = decodeURIComponent(window.location.search)
48       .substring(1)
49       .split("confirmation_token=")[1];
50     this.attemptLogin({ token, ...this.loginCreds })
51       .then(res => {
52         this.transferToDashboard();
53         console.log(res);
54       })
55       .catch(err => {
56         console.log(err);
57       });
58   },
59 }
60 }
61 </script>
62
63
```



Reward

DATE

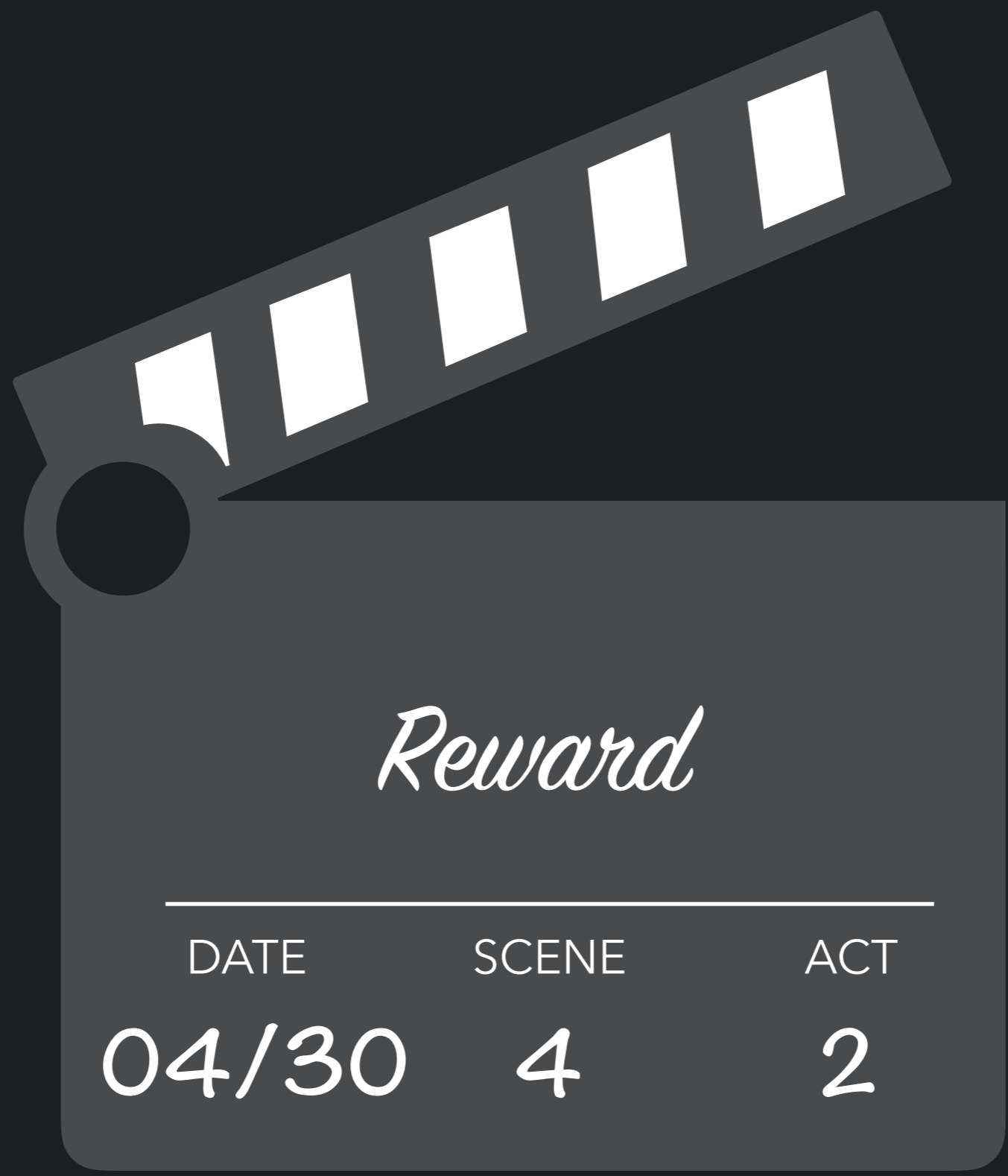
SCENE

ACT

04/30

4

2



Reward

DATE	SCENE	ACT
04/30	4	2



Reward

DATE

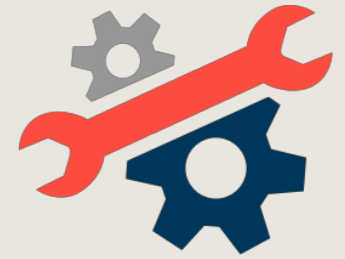
SCENE

ACT

04/30

4

2



Identity

Identity Enterprise

API endpoint: <https://chipie.netlify.com/.netlify/identity>

[Learn more in the documentation](#) ↗

⚙ Settings and usage

Identity audit log

🔍 Search users

Invite users

H hello@shortdiv.com

visitor

Created on Apr 26 (4 days ago)



D divya@netlify.com

editor

Created on Apr 23 (7 days ago)



Password????



Act 3

Return





The Road Back

DATE

SCENE

ACT

04/30

1

3



The Road Back

DATE	SCENE	ACT
04/30	1	3



The Road Back

DATE

SCENE

ACT

04/30

1

3



There is no cloud, just someone else's server



*Resurrection/
Return with Elixir*

DATE

SCENE

ACT

04/30

2

3



*Resurrection/
Return with Elixir*

DATE	SCENE	ACT
04/30	2	3



*Resurrection/
Return with Elixir*

DATE

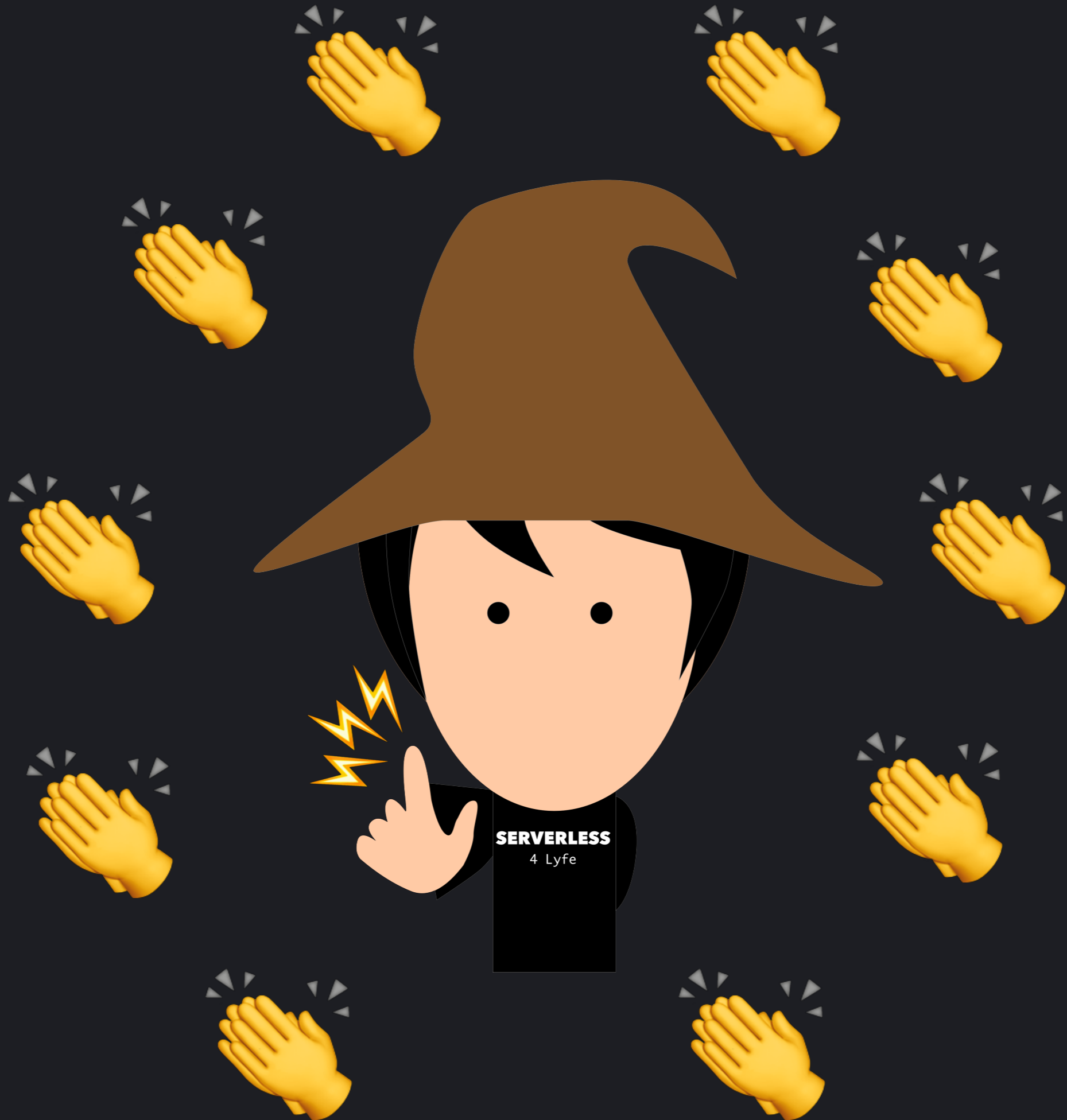
SCENE

ACT

04/30

2

3



SERVERLESS

4 Lyfe

chipie.netlify.live



github.com/shortdiv/chipie

Divya Sasidharan



netlify

@shortdiv

goto;
chicago



Please

**Remember to
rate this session**

Thank you!

