

# DLTI-db

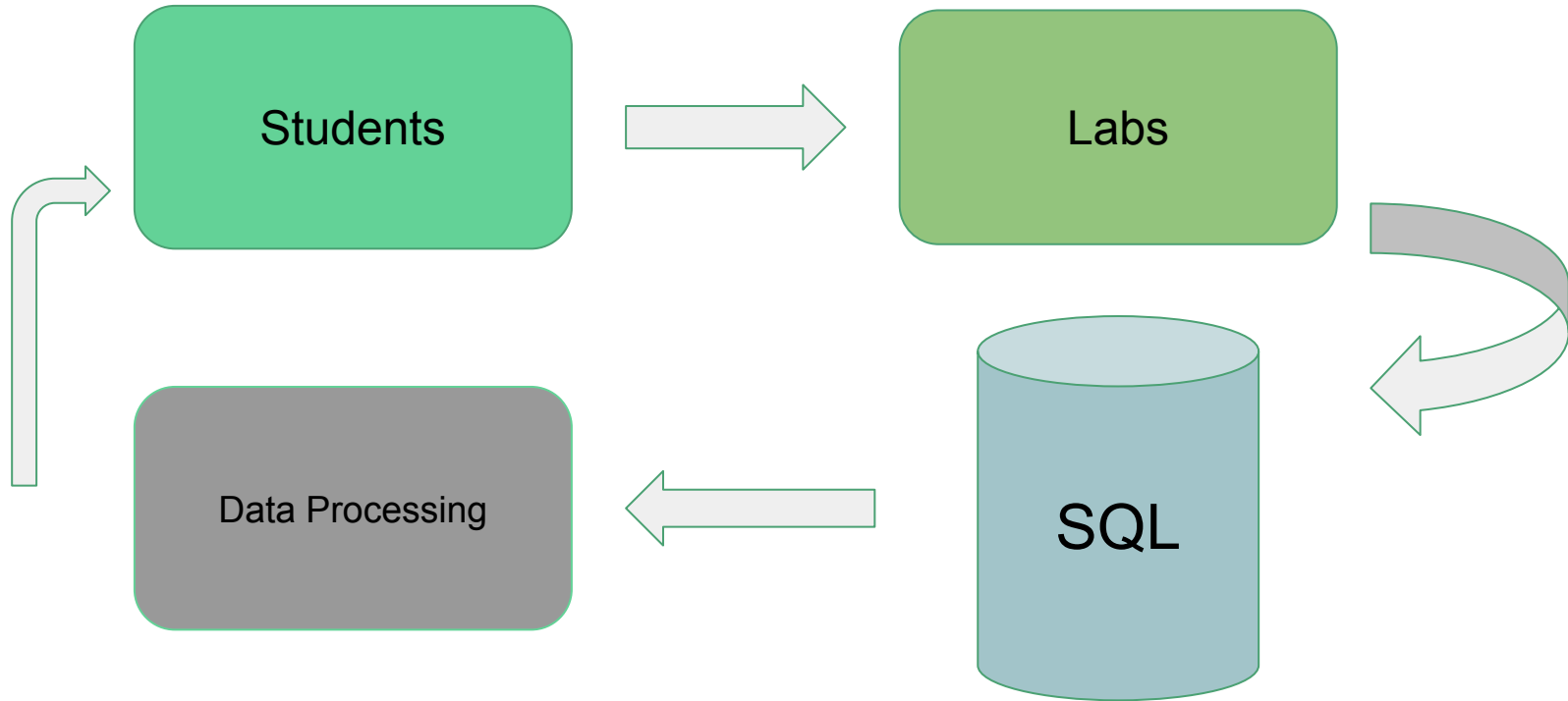
---

William Wu, Will Smith

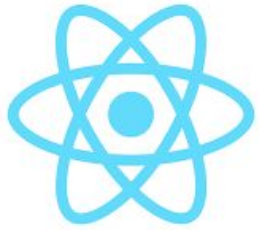
# What's the problem?

- Unable to grab the state from the database and restore it back to the GUI
- Not all filter controls exist, some don't update properly
  - sliders impacting all graphs are slow to update

# What we want



# Tech Stack



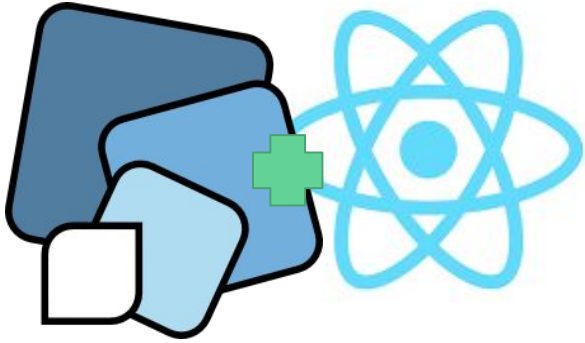
# Labor Division

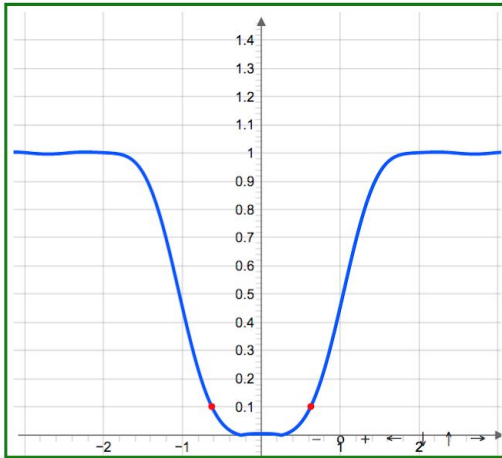
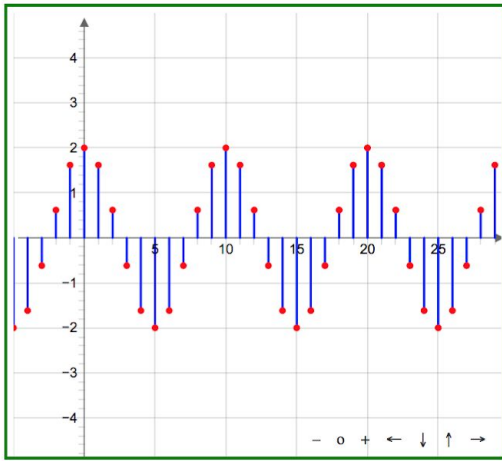
Will - GUI, React

William - SQL, Node/Express

Front end

# Tools





Controls

Hamming-Sinc Hig... ▾

**Amplitude**

Frequency (\* 2pi)

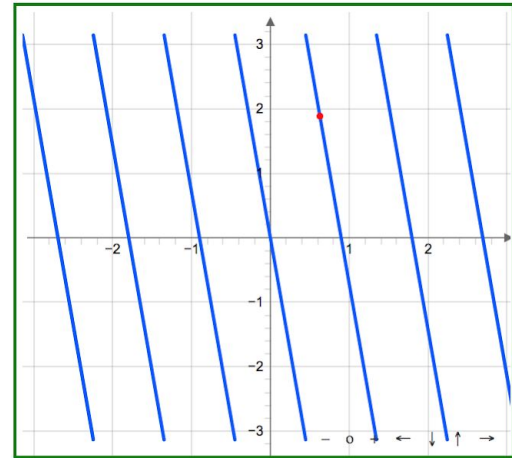
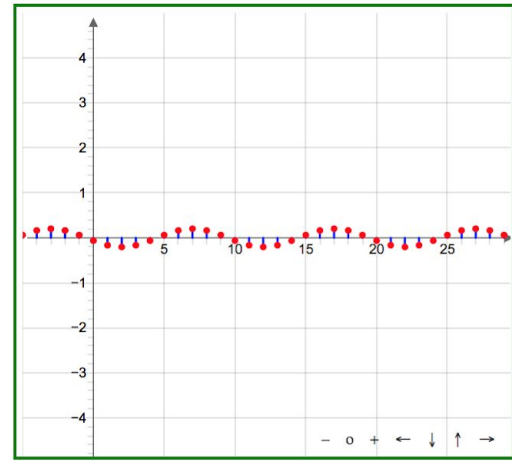
Phase (\* pi)

DC Level

**HAMMING-SINC HIGH-PASS**

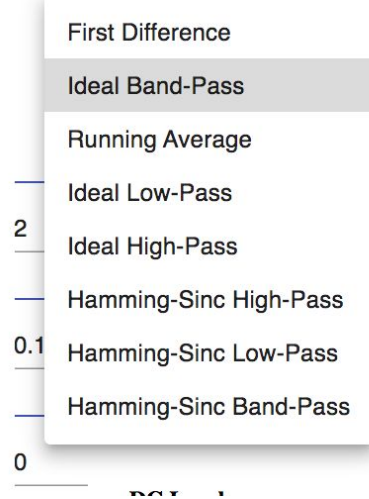
Cutoff Freq

Length





```
switch(select) {
```



## IDEAL BAND-PASS

Cutoff Freq 1



0.628

Cutoff Freq 2



2.199

Phase Slope



```
case 2:  
  return (<div><strong>IDEAL BAND-PASS</strong><p> </p>  
    <strong> Cutoff Freq 1 </strong>  
    <SliderIn value={this.state.idealBands[0]} sliderChange={(e, v) => {  
      this.setState( state: {idealBands: [v, this.state.idealBands[1]]});  
      this.state.idealBands[0] = v;  
      this.updateGraphs();  
      this.updateMagGraph();  
      this.updatePhaseGraph();  
    }} step={.2}
```

Instantiate

```
this.outStems =  
brd.create('curve', [[], []], {strokeWidth:2,  
color:"blue", highlightStrokeColor:"blue"});
```

Update

```
this.inStems.dataX = [];  
this.inStems.dataY = [];  
this.inStems.updateDataArray = () =>
```

Back end

# Database

- MySQL Database
  - Takes student\_id, Lab\_id, and Question\_number
  - Seven general variables for Amp, Freq, Phase, DC, and filter controls
- Ended up making ONE table with a primary key
  - Prevents duplicate entries

Student_id	Lab_id	Question_number	Variable1	Variable2	Variable3	Variable4	Variable5	Variable6	Variable7
1234	1	1	2.5	0.28	0.29	0.3	10	11	2
90134325	1	1	1.48	0.1	-0.23	-0.75	12	11	2
90324199	1	1	2.99	0.24	-0.02	0.46	8	11	1
910432643	1	1	1.48	0.1	-0.23	-0.75	2	11	3

# API

```
var con = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: "password",
  database: 'labs',
});
```

- Server.js
- Node.js/Express/Cors
- App.get() and App.post() methods

```
handleSubmit = (e) => {
```

```
handleGetData = (e) => {
```

Submit

Get state

DEMO

# Future Implementations

- Frontend additions
  - Mathjax and equations on top of the GUIs
  - Cleaner UI
    - shift organization
- Backend additions
  - Put the GUI on VM server
  - Creating a template of the API for future GUIs to use
- Move onto host machine

# Reflection

- Challenges
  - Teammate dropping
  - Frontend: more controls than original demo, varying controls & parameters per filter
  - Backend: Getting some of the routing issues solved. Covid-19.
- Accomplishments
  - DLTI lab demo is moving forward in terms of getting data/answers from students
  - Students will now be able to grab their answers back from the database
  - Looking into a template for the REST API to fit in with future GUIs



# THANK YOU

Questions?