

# INTELLIGENT REVIEW SYSTEM

BACK END TEAM:  
ANTONIO GONCALVES  
PETER NGUYEN  
WINAGODWIN ANYANWU JR.

**CREATING THE NEXT®**

## Introductions

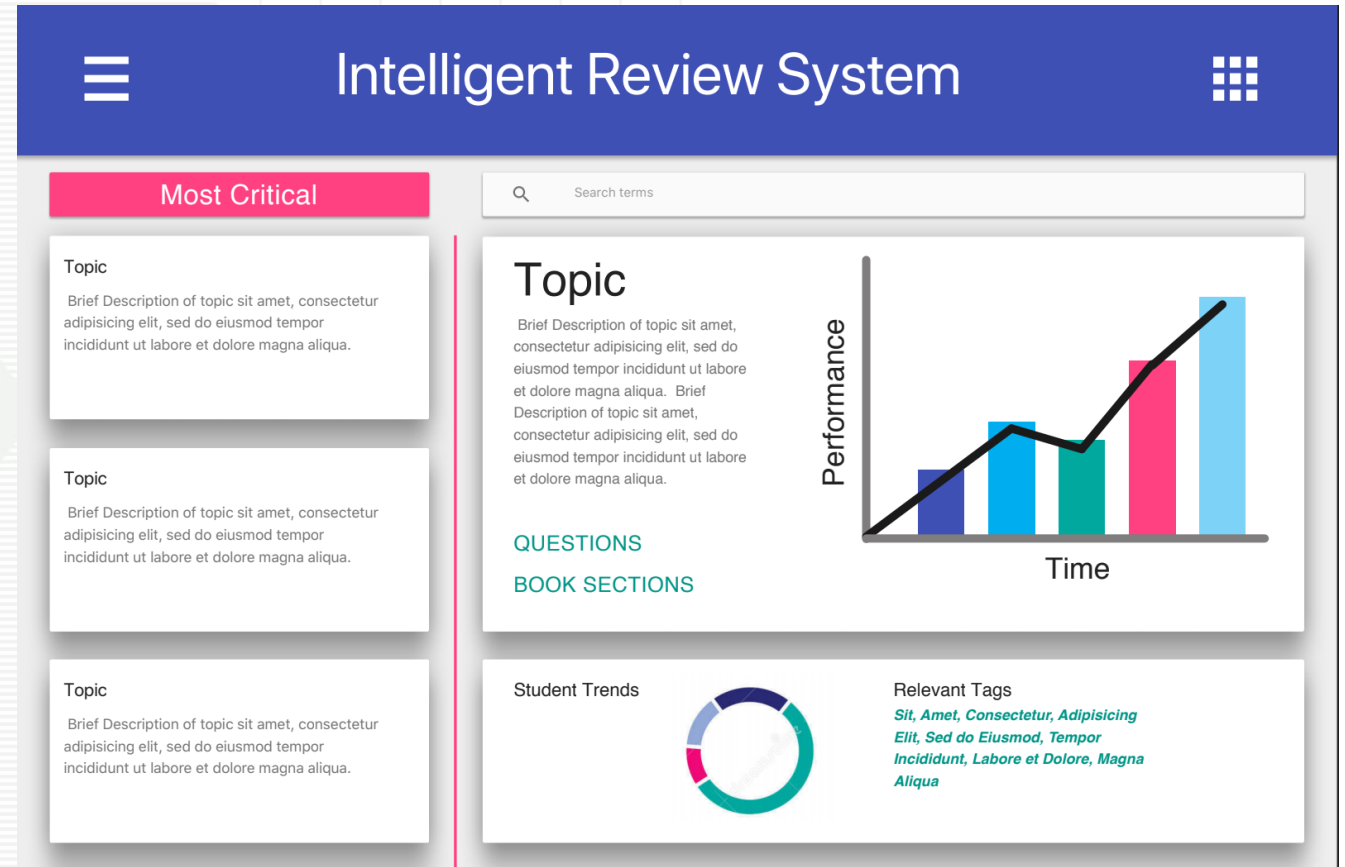
- **Antonio Gonclaves**
  - 4th Year Mechanical Engineering Major, CS Minor, Finance Certificate
  - Experience: Java, Python, MatLab
- **Peter Nguyen**
  - 3rd Year Computer Engineering Major
  - Experience: C/C++, Python(scikit-learn machine learning algorithms), Matlab
- **Winagodwin Anyanwu Jr.**
  - 3rd Year Computer Science Major
  - Experience: Java, C, Android, Python

## OBJECTIVE

CREATE A PLATFORM FOR TA'S TO IDENTIFY  
AND REVIEW TOPICS THAT STUDENTS  
ARE STRUGGLING WITH THE MOST.

# Overview

- Approach
- Setup
- Research and Process
- Linkage
- Future Implementations
- What we have learned

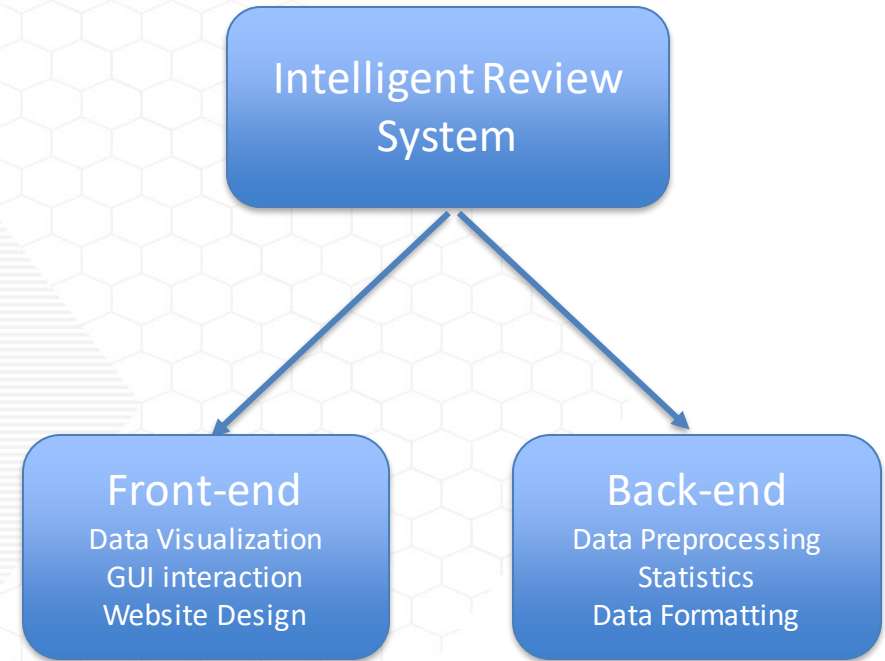


The screenshot shows the 'Intelligent Review System' interface. It features a blue header with a hamburger menu icon on the left and a grid icon on the right. Below the header, there is a search bar with the placeholder text 'Search terms'. The main content area is divided into several sections:

- Most Critical:** A pink header section containing three topic cards. Each card has a 'Topic' label and a brief description: 'Brief Description of topic sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.'
- Topic:** A larger section with a 'Topic' label and a detailed description: 'Brief Description of topic sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Brief Description of topic sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.'
- Performance:** A bar chart showing performance over time. The y-axis is labeled 'Performance' and the x-axis is labeled 'Time'. There are five bars of increasing height, with a black line graph overlaid showing an upward trend.
- QUESTIONS** and **BOOK SECTIONS:** Two blue text labels.
- Student Trends:** A section with a circular progress indicator and the text 'Student Trends'.
- Relevant Tags:** A section with the text 'Relevant Tags' and a list of tags: 'Sit, Amet, Consectetur, Adipiscing Elit, Sed do Eiusmod, Tempor Incidunt, Labore et Dolore, Magna Aliqua'.

## Approach

- Add direct linkage from the database to front-end.
- Research on past projects.
- The back-end team used Python to mine data.
- Back-end produce usable json files for the front-end team.



## Infrastructure Setup

- Initially looked at past resources from ITS.
  - Problematic due to outdated virtual machines and documentation.
- Ultimately, setup our own virtual machine which proved lengthy.
  - Every step along the process had its own problem.



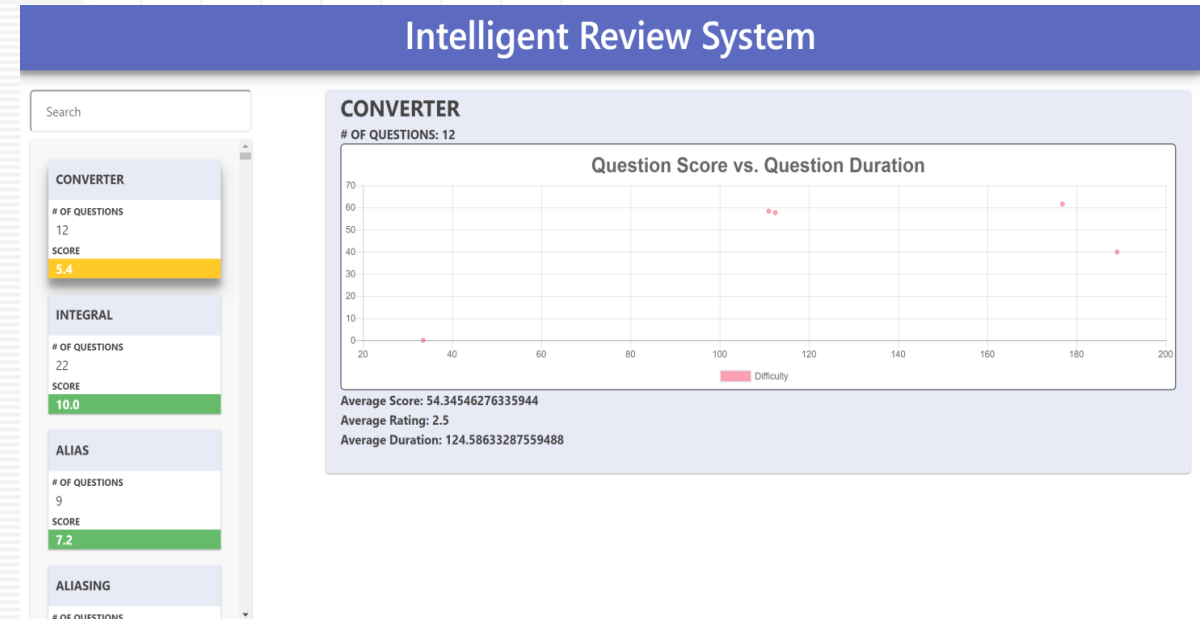
ubuntu 

MySQL™



## Process

- The team was able to successfully send data from the database to the front-end.
- The website can display a variety of statistics.



# Linkage

```
mysql> SELECT * from stats_3503;
```

id	question_id	concept_id	current_chapter	answered	score	rating	comment	tags	epochtime	duration	event
1	385	NULL	3	D	100	2	NULL	NULL	1453788828	197	chapter
2	653	NULL	3	B	0	4	NULL	NULL	1453789042	240	chapter
3	3543	NULL	3	NULL	NULL	NULL	NULL	NULL	1453789322	NULL	equation
4	3543	NULL	3	NULL	NULL	NULL	NULL	NULL	1453789326	NULL	equation

```
def getUserData(connectx, user_id_list):  
  
    stat_cursor = connectx.cursor(buffered=True)  
    user_dict = {}  
  
    for user_id in user_id_list:  
        user_dict[user_id] = []  
        try:  
            stats_query = ("SELECT * FROM stats_{0:d} WHERE score IS NOT NULL".format(user_id))  
            stat_cursor.execute(stats_query)  
            for stat in stat_cursor:  
                question_id = stat[1]  
                score = stat[5]  
                timestamp = stat[9]  
                assignment = stat[3]  
                duration = stat[10];  
                rating = stat[6]  
                if type(assignment) != int:  
                    assignment = 0  
                user_dict[user_id].append( {'q_id':question_id, 'score':score, 'ts':timestamp,  
                    'assignment':assignment, 'duration':duration, 'rating':rating} )  
        except:  
            print("Table for user {0:d} does not exist".format(user_id))  
  
    stat_cursor.close()  
    return user_dict
```

```
"['AM']": [  
    3428,  
    3327,  
    3303,  
    867,  
    349,  
    414,  
    241,  
    517  
],  
"['ASCII']": [  
    55  
],  
"['BIBO']": [  
    2165,  
    3217,  
    3283  
],  
"['Blackman']": [  
    206,  
    207,  
    208,  
    209,  
    767  
],  
"['C-to-D']": [  
    3315,  
    3314  
],  
"['D-to-A']": [  
    1051,  
    278,  
    217,  
    9,  
    5  
],  
1.
```



# Linkage (continued)

```
{  
  "Topics": [  
    {  
      "name": "converter",  
      "Average Score": 54.34546276335944,  
      "Average Rating": 2.5,  
      "Average Duration": 124.58633287559488,  
      "difficulty": null,  
      "questions": [  
        505,  
        3435,  
        3370,  
        3366,  
        3360,  
        3315,  
        3387,  
        1205,  
        1084,  
        775,  
        676,  
        3314  
      ]  
    },  
  ],  
}
```

### Intelligent Review System

Search

CONVERTER
# OF QUESTIONS: 12
SCORE: 5.4

INTEGRAL

# OF QUESTIONS: 22
SCORE: 10.0

ALIAS

# OF QUESTIONS: 9
SCORE: 7.2

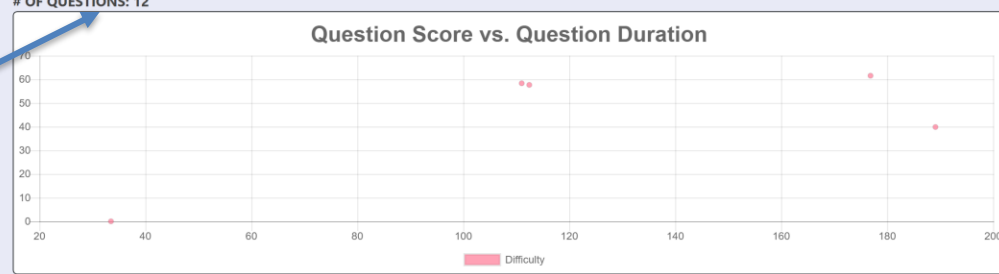
ALIASING

# OF QUESTIONS: 9
-------------------

**CONVERTER**

# OF QUESTIONS: 12

Question Score vs. Question Duration



Average Score: 54.34546276335944  
Average Rating: 2.5  
Average Duration: 124.58633287559488

## Future Implementations

- Machine Learning implementation
- Automatic updates
- Future VIP teams would be able to use our code to continue

## What We Learned

- Virtual Machine Setup
- Python and SQL connections
- Data processing
- Formatting Data

# INTELLIGENT REVIEW SYSTEM

FRONT-END TEAM:  
NEAL KURANDE  
LUCAS PHILLIPS

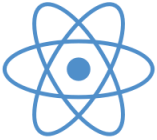
**CREATING THE NEXT®**

## Introductions

- Neal Kurande
  - 3rd year Computer Engineer
  - Experience: C/C++, Python, Java, JavaScript, MATLAB
- Lucas Phillips
  - 5th year CS major (Spanish minor)
  - Experience: HTML/CSS, JavaScript, ReactJS/React-native, node, PHP, MySQL, Python, Java, C, Assembly

# Infrastructure Setup

Create React App



Official. No Setup. Minimal.



React



HEROKU



yarn


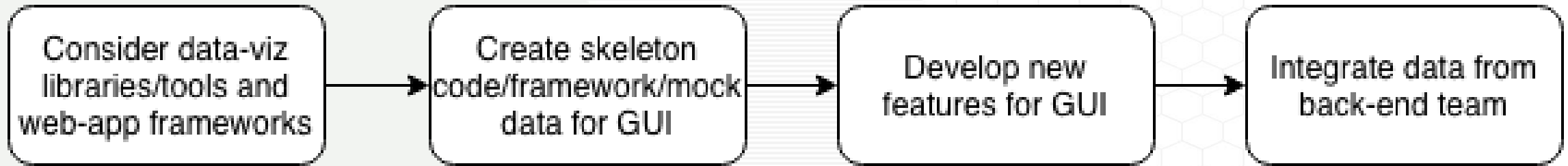


Chart.js



# Approach



# Initial Setup



## Intelligent Review System

Topic Name This is a description of the topic

Topic Name 2 This is a description of the topic 2



# Card Communication

## Intelligent Review System

Search

**Fourier Transform**

DESCRIPTION  
This is a description of the topic

DIFFICULTY  
**9**

**Radian Frequency**

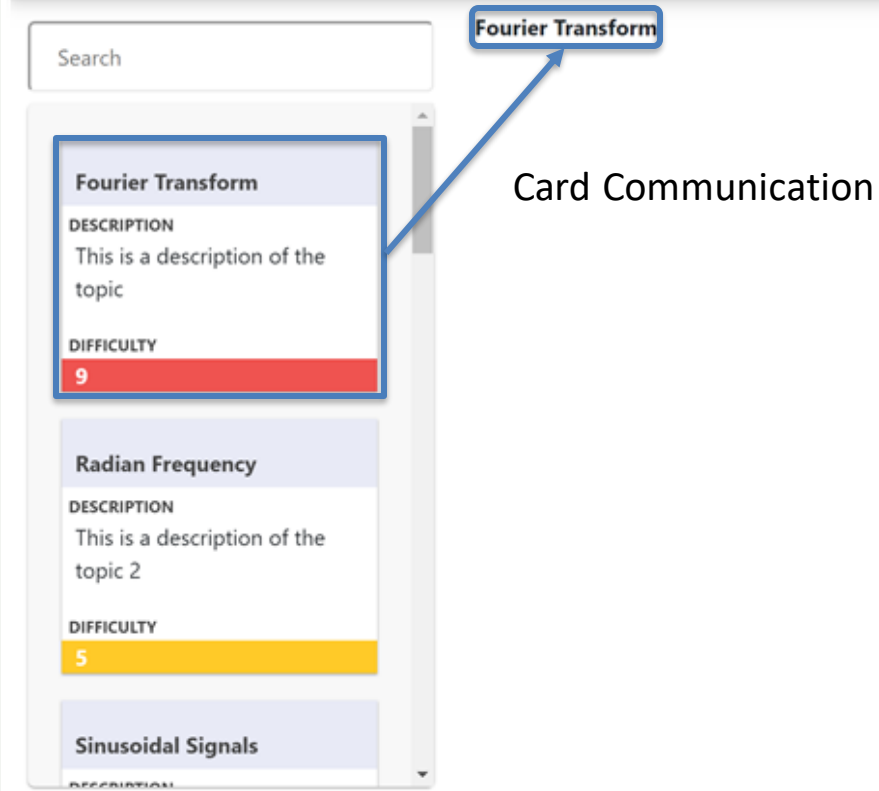
DESCRIPTION  
This is a description of the topic 2

DIFFICULTY  
**5**

**Sinusoidal Signals**

DESCRIPTION

Card Communication



## Intelligent Review System

Radian Frequency

Radian

Radian Frequency

DESCRIPTION  
This is a description of the topic

DIFFICULTY  
5

Search bar filters to the right topic

```
15  handleChange(event) {  
16    let query = event.target.value.toLowerCase();  
17    let displayedTopics = this.props.data.topics.filter((t) => {  
18      let searchValue = t.name.toLowerCase();  
19      return searchValue.indexOf(query) !== -1;  
20    })  
21  
22    this.setState({topics: displayedTopics});  
23  }  
24
```

# Chart Linkage with Chart.js

## Intelligent Review System

Search

**Fourier Transform**

DESCRIPTION  
This is a description of the topic

DIFFICULTY  
**9**

**Radian Frequency**

DESCRIPTION  
This is a description of the topic

DIFFICULTY  
**5**

**Sinusoidal Signals**

DESCRIPTION  
This is a description of the topic

DIFFICULTY

### Fourier Transform

# of Questions: 5



```
1 import React, {Component} from 'react';
2 import { Bar, Scatter } from 'react-chartjs-2';
3
4 class Chart extends Component{
5   constructor(props){
6     super(props);
7     this.state = {
8       chartData: props.chartData
9     }
10  }
```

## Intelligent Review System

Search

SCORE  
10.0

ALIAS

# OF QUESTIONS  
9

SCORE  
7.2

ALIASING

# OF QUESTIONS  
42

SCORE  
4.8

FOLDING

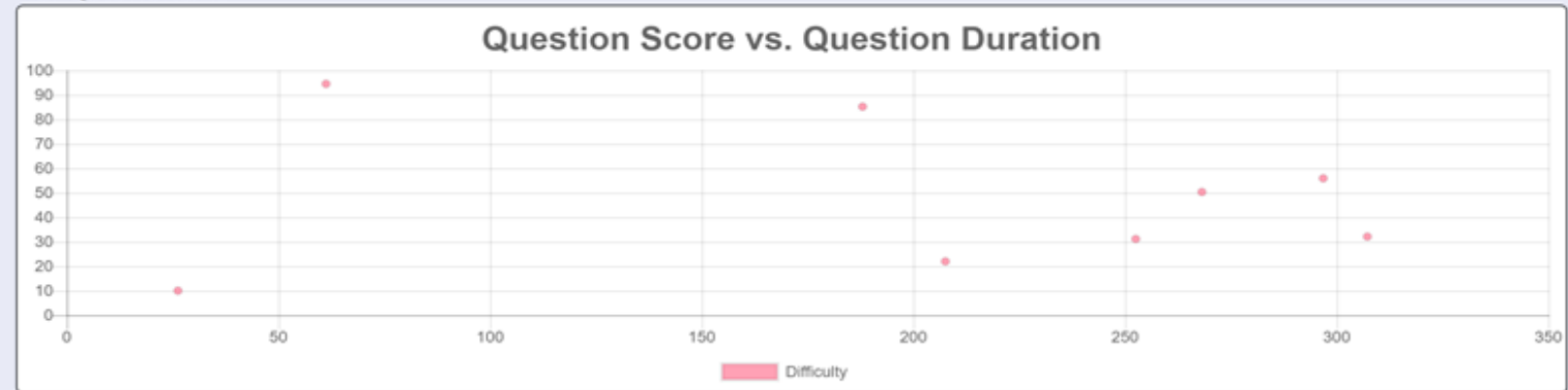
# OF QUESTIONS  
10

SCORE  
0.0

AMPLITUDE

### ALIASING

# OF QUESTIONS: 42



```
render() {  
  return (  
    <div className="App">  
      <Header className="header"/>  
      <div className="container-fluid">  
        <div className="row">  
          <div className="col-sm-3">  
            <TopicList className="topic-list" data={data} changeTopic={this.changeTopic}/>  
          </div>  
          <div className="col-lg-9">  
            <TopicInfo className="topicInfo" topic={this.state.currentTopic}/>  
          </div>  
        </div>  
      </div>  
    </div>  
  )  
}
```

## Next Steps

- Filter(s)
- Realtime integration with back-end
- Compare/more information

Link to Demo



<https://intelligent-review-system.herokuapp.com>

**QUESTIONS?**

**CREATING THE NEXT®**