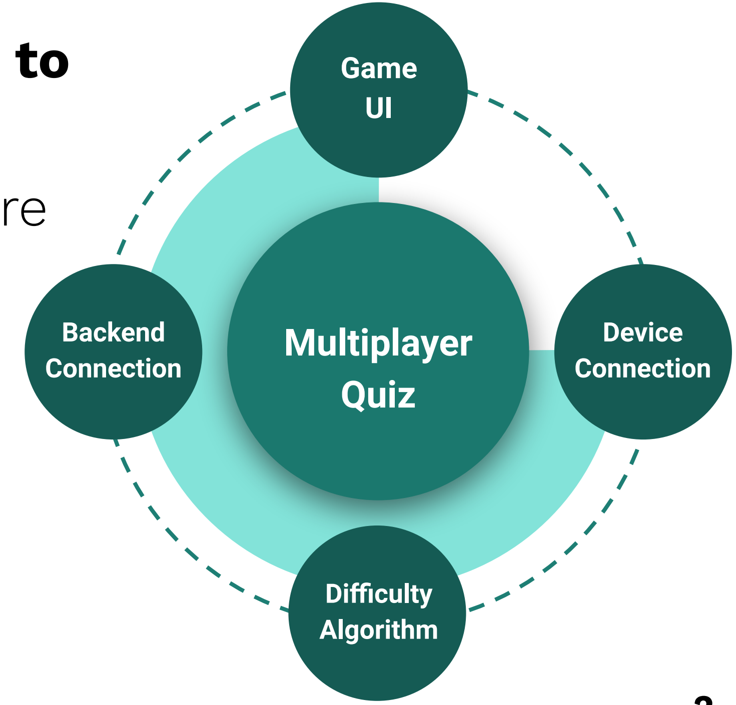
Multiplayer Quiz App

Alex Tang, Alyssa Zhu, Nirjhar Deb, Dongzhao Song,
Xinyi Wang, Justin Kim

Motivation and Goals

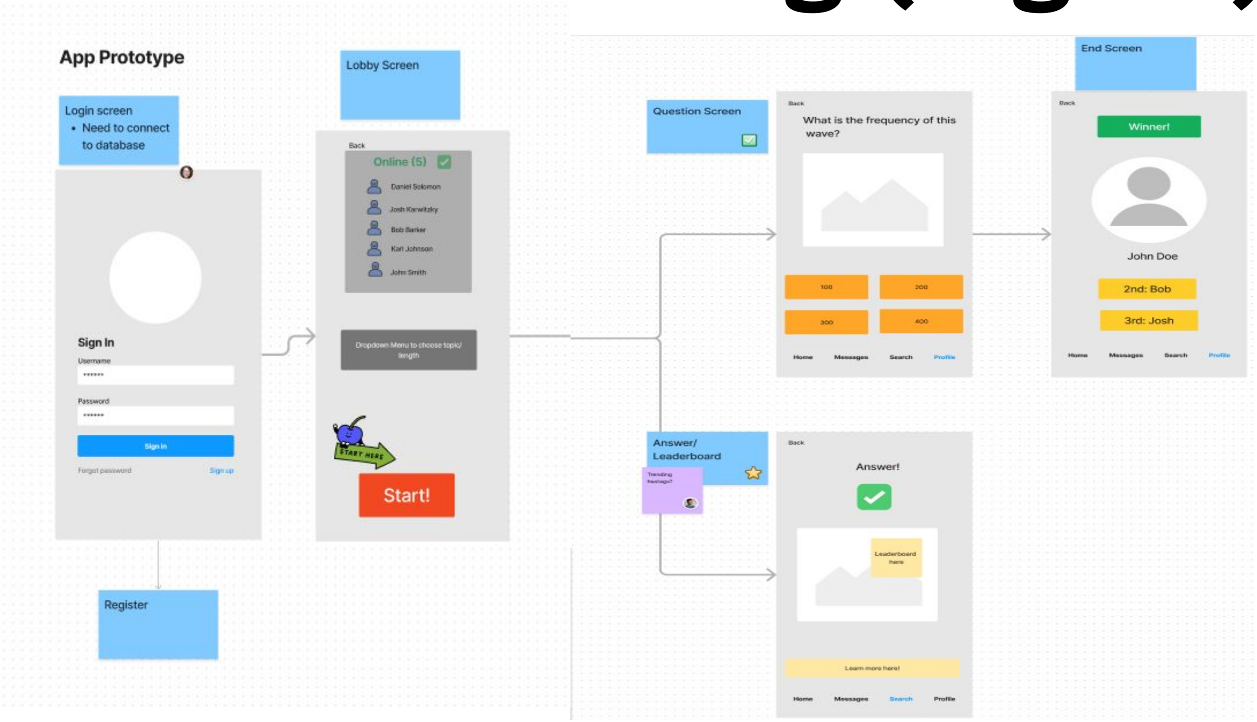
- **Create competitive environment to promote learning**
  - □ Add live multiplayer quiz feature
  - □ Develop game user interface
  - □ Multi-device connection
  - □ Connect to backend
  - □ Create question difficulty/selection algorithm



Frontend (Game UI)

Alyssa and Alex

Initial Planning (Figma)



App Prototype

Login screen
- Need to connect to database

Lobby Screen

Question Screen

What is the frequency of this wave?

Answer/ Leaderboard

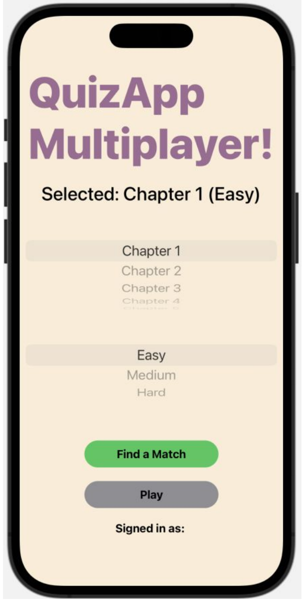
Trending hashtags?

End Screen

Register

Screens

More questions left



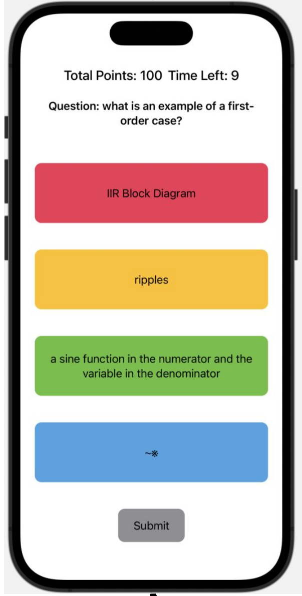
QuizApp Multiplayer!

Selected: Chapter 1 (Easy)

Chapter 1
Chapter 2
Chapter 3
Chapter 4

Easy
Medium
Hard

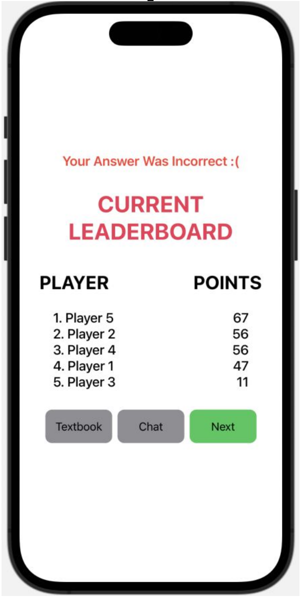
Signed in as:



Total Points: 100 Time Left: 9

Question: what is an example of a first-order case?

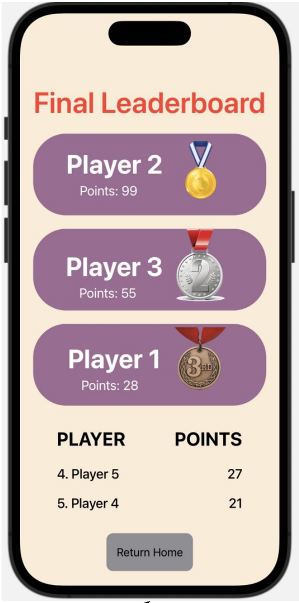
IIR Block Diagram
ripples
a sine function in the numerator and the variable in the denominator
~⁎
Submit



Your Answer Was Incorrect :(

CURRENT LEADERBOARD

PLAYER	POINTS
1. Player 5	67
2. Player 2	56
3. Player 4	56
4. Player 1	47
5. Player 3	11



Final Leaderboard

Player 2
Points: 99

Player 3
Points: 55

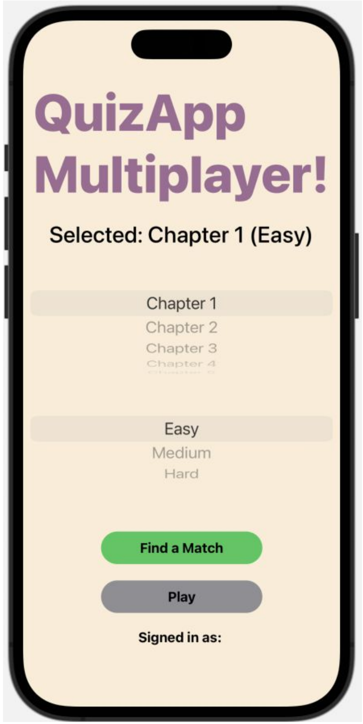
Player 1
Points: 28

PLAYER	POINTS
4. Player 5	27
5. Player 4	21

End of game

LobbyView

- User can select chapter and difficulty to pull questions from
- Game options will be located here
- Find match button connects to GameCenter

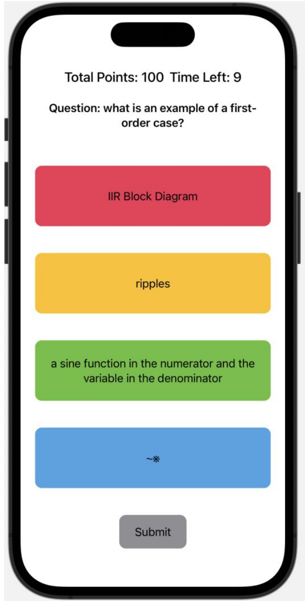


QuizApp Multiplayer!

Selected: Chapter 1 (Easy)

Signed in as:

QuestionView/Leaderboard

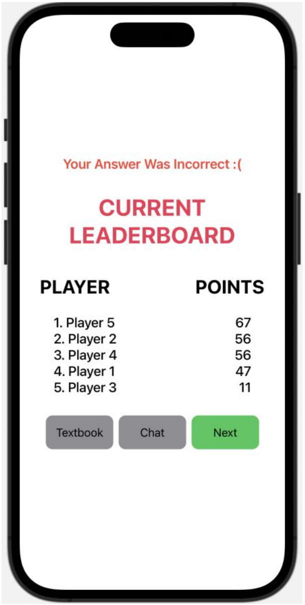


Total Points: 100 Time Left: 9

Question: what is an example of a first-order case?

IIR Block Diagram
ripples
a sine function in the numerator and the variable in the denominator
~※
Submit

- Pulls a random set of questions and answers from the json file
- Provides feedback and updates an array containing players scores
- Two leaderboard screens (transitional and final) to show player scores at any point



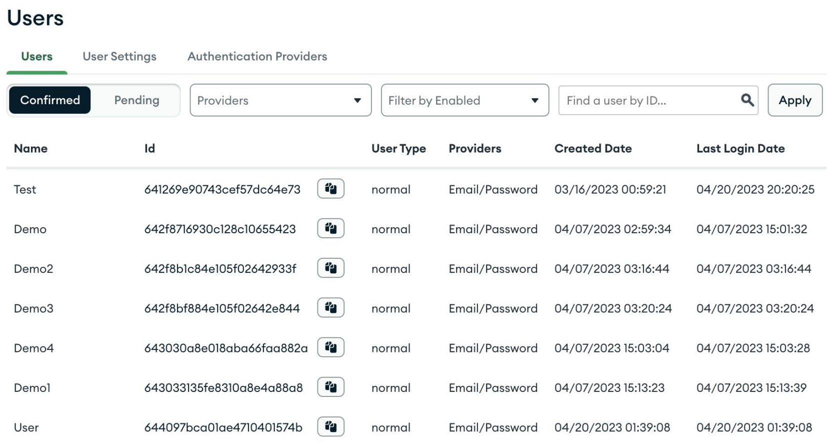
Your Answer Was Incorrect :(

CURRENT LEADERBOARD

PLAYER	POINTS
1. Player 5	67
2. Player 2	56
3. Player 4	56
4. Player 1	47
5. Player 3	11

Login/Registration

- MongoDB App Service
  - User authentication
  - Database
- Connect using Realm
- Takes in and records registration input for future logins

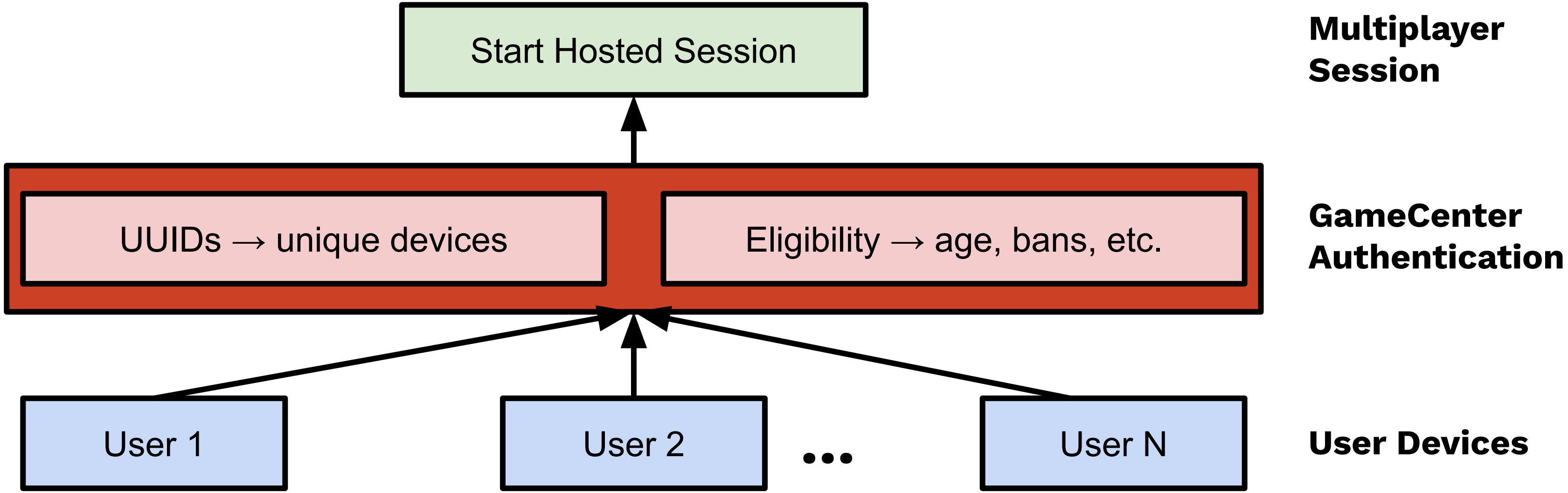


Name	Id	User Type	Providers	Created Date	Last Login Date
Test	641269e90743cef57dc64e73	normal	Email/Password	03/16/2023 00:59:21	04/20/2023 20:20:25
Demo	642f8716930c128c10655423	normal	Email/Password	04/07/2023 02:59:34	04/07/2023 15:01:32
Demo2	642f8b1c84e105f02642933f	normal	Email/Password	04/07/2023 03:16:44	04/07/2023 03:16:44
Demo3	642f8bf884e105f02642e844	normal	Email/Password	04/07/2023 03:20:24	04/07/2023 03:20:24
Demo4	643030a8e018aba66faa882a	normal	Email/Password	04/07/2023 15:03:04	04/07/2023 15:03:28
Demo1	643033135fe8310a8e4a88a8	normal	Email/Password	04/07/2023 15:13:23	04/07/2023 15:13:39
User	644097bca01ae4710401574b	normal	Email/Password	04/20/2023 01:39:08	04/20/2023 01:39:08

Multiplayer → Connect the World…

Nirjhar

Multi-device Connection



Start Hosted Session

UUIDs → unique devices

Eligibility → age, bans, etc.

User 1

User 2

...

User N

Multiplayer Session

GameCenter Authentication

User Devices

Why GameCenter?

- Offloads networking and hosting responsibility to Apple
- Low latency
- Unlimited amount of users

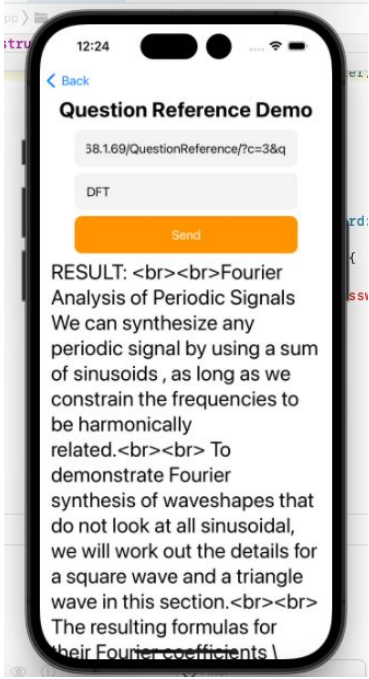
 Developer

- **Caveat:** $99/yr Apple Developer subscription 😈

Frontend (Algorithm)

Xinyi

Screens



12:24

Back

Question Reference Demo

58.1.69/QuestionReference/?c=3&q

DFT

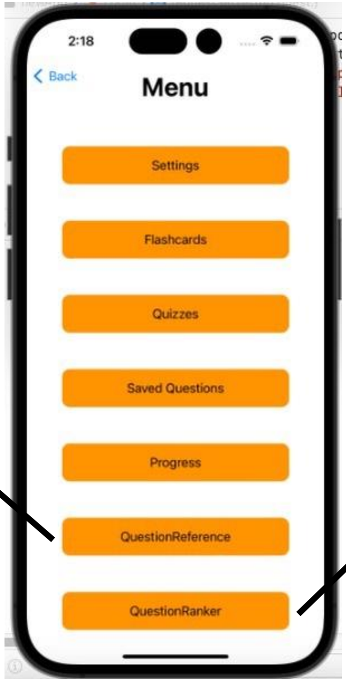
Send

RESULT:

Fourier Analysis of Periodic Signals We can synthesize any periodic signal by using a sum of sinusoids , as long as we constrain the frequencies to be harmonically related.

 To demonstrate Fourier synthesis of waveshapes that do not look at all sinusoidal, we will work out the details for a square wave and a triangle wave in this section.

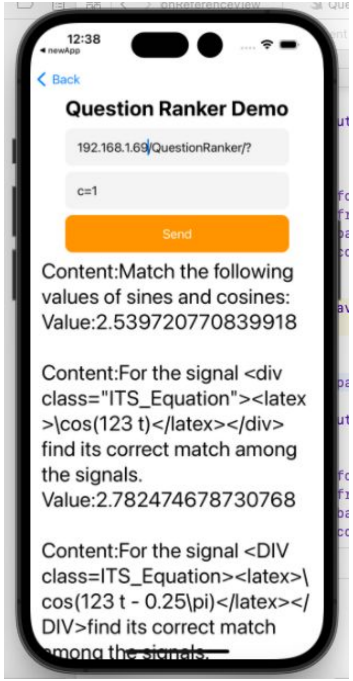
 The resulting formulas for their Fourier coefficients \



2:18

Back

Menu



12:38

◂ newApp

Back

Question Ranker Demo

192.168.1.69/QuestionRanker/?

c=1

Send

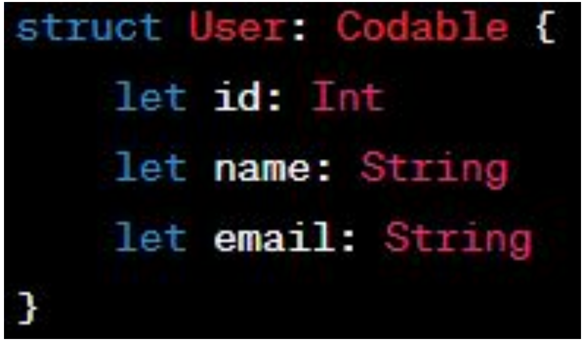
Content:Match the following values of sines and cosines: Value:2.539720770839918

Content:For the signal <div class="ITS_Equation"><latex >\cos(123 t)</latex></div> find its correct match among the signals. Value:2.782474678730768

Content:For the signal <DIV class=ITS_Equation><latex>\cos(123 t - 0.25\pi)</latex></DIV>find its correct match among the signals.

Restful URL API Call

- Define a struct to represent the JSON response data
- Create URL, URLSession, and URLRequest object, data task to perform request
- Decode the response data into the format you need using JASONDecoder



```
struct User: Codable {
    let id: Int
    let name: String
    let email: String
}
```

Restful Server

Dongzhao

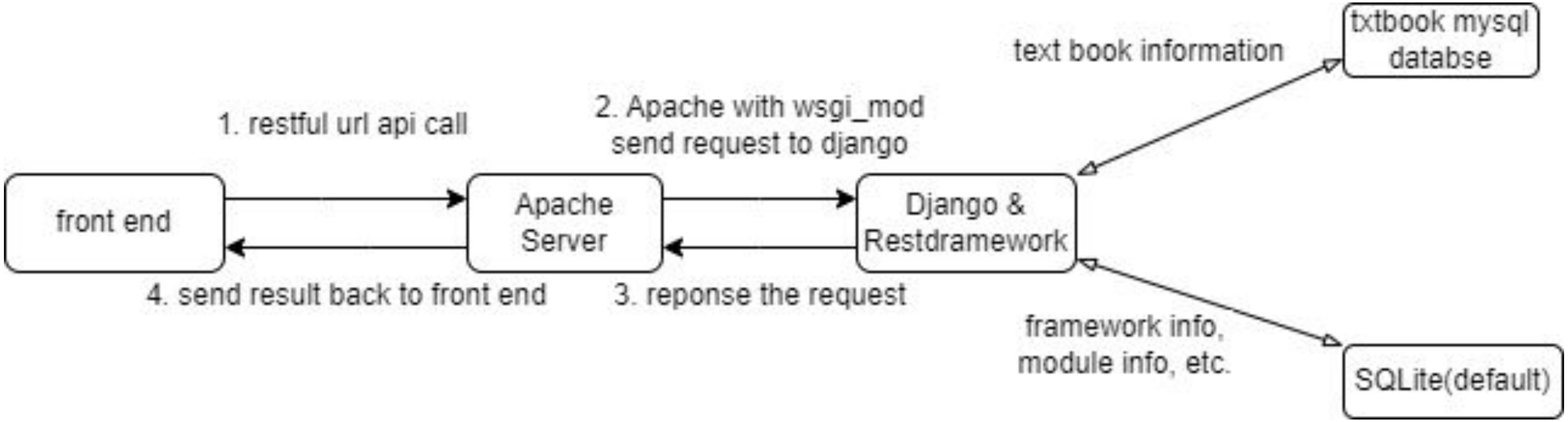
Server

Django - A very popular python server development framework. It is a MVC (model, view, controller) framework. Itself support web page and identification. We only used small part of its ability.

Restframework - A restful api framework based on Django. It provides convenient method such as serializer to get JSON from Django models.

Question Reference & Question Ranking - the algorithms we wrote in this and last semester. One supposes to response a textbook section related to an input question. The other, with an chapter number as input, will give a list of ranked questions based on their hardness.

Server



text book information

txtbook mysql databse

1. restful url api call

2. Apache with wsgi_mod send request to django

front end

Apache Server

Django & Restdramework

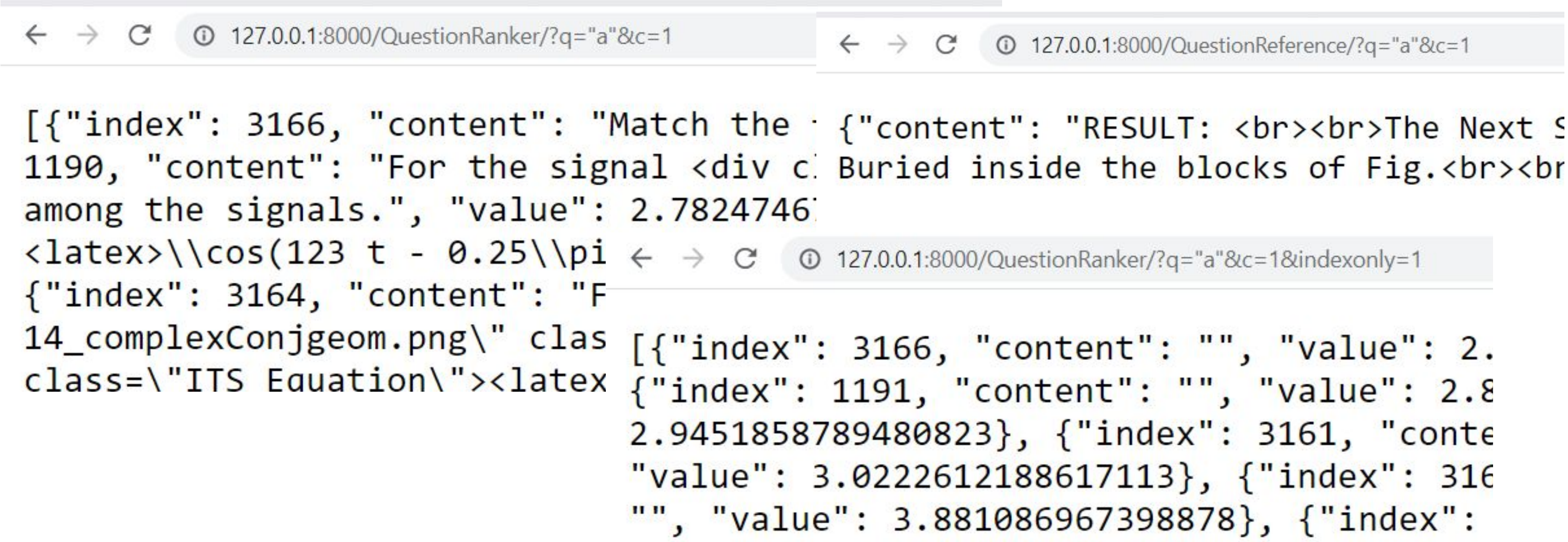
4. send result back to front end

3. reponse the request

framework info, module info, etc.

SQLite(default)

API Response



127.0.0.1:8000/QuestionRanker/?q="a"&c=1

```
[{"index": 3166, "content": "Match the
1190, "content": "For the signal <div cl
among the signals.", "value": 2.7824746
<latex>\\cos(123 t - 0.25\\pi
{"index": 3164, "content": "F
14_complexConjgeom.png\" clas
class=\"ITS Equation\"><latex
```

127.0.0.1:8000/QuestionReference/?q="a"&c=1

```
{"content": "RESULT: <br><br>The Next S
Buried inside the blocks of Fig.<br><br
```

127.0.0.1:8000/QuestionRanker/?q="a"&c=1&indexonly=1

```
[{"index": 3166, "content": "", "value": 2.
{"index": 1191, "content": "", "value": 2.8
2.9451858789480823}, {"index": 3161, "conte
"value": 3.0222612188617113}, {"index": 316
"", "value": 3.881086967398878}, {"index":
```

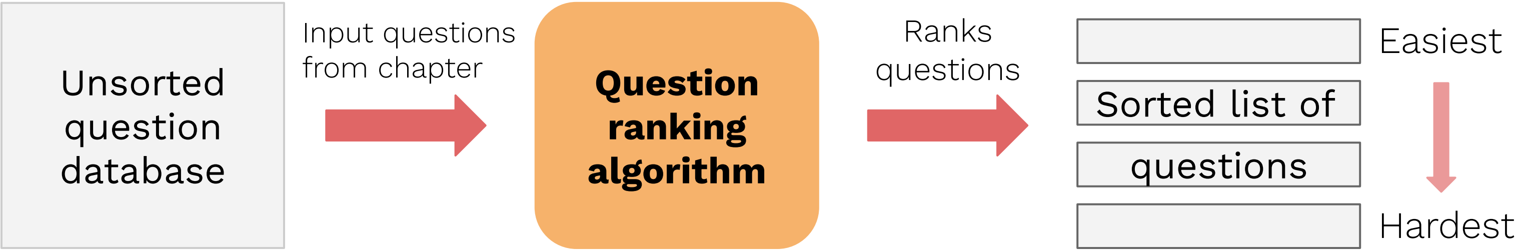
Algorithm Implementation

Justin

Question Ranking

Question ranking algorithm - Given a question, estimate its difficulty with a score that can be compared to those of other questions

Motivation: Using this algorithm in question selection would allow QuizApp users to cover foundational topics first before moving on to more advanced topics



Unsorted question database

Input questions from chapter

Question ranking algorithm

Ranks questions

Sorted list of questions

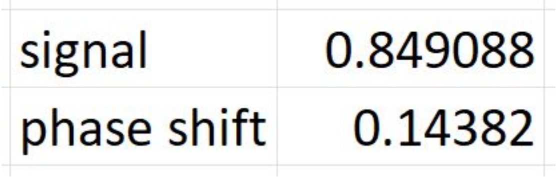
Easiest

Hardest

Keywords/Calculation

We used MonkeyLearn to extract keywords and their relevancy score from the textbook.

Keyword -->



signal	0.849088
phase shift	0.14382

<-- Relevancy score, **r**
(Higher score indicates higher relevancy within text)

We use the value (1-**r**) to find the difficulty score **d** of a component using the weighted sum of its hardest 3 keywords:

$$\mathbf{d} = 0.6*(1-\mathbf{r}_1) + 0.25*(1-\mathbf{r}_2) + 0.15*(1-\mathbf{r}_3)$$

Then, we calculate the final difficulty index of a **question**/**answer** as:

$$\mathbf{D} = \mathbf{d}_{\text{answer}} + \ln(\text{length of answer}) + \mathbf{d}_{\text{question}} + \ln(\text{length of question})$$

Future Improvements

Future Improvements

- Connecting the questions to database
- Utilizing question difficulty algorithm for question selection
- Add authentication to backend if need
- Adding a tutor role, who can send hints to students via GameCenter's multi-user information transfer

Demos