

# Tutor JS Frontend



VIP Intelligent Tutoring Systems Spring 2021

# The Team

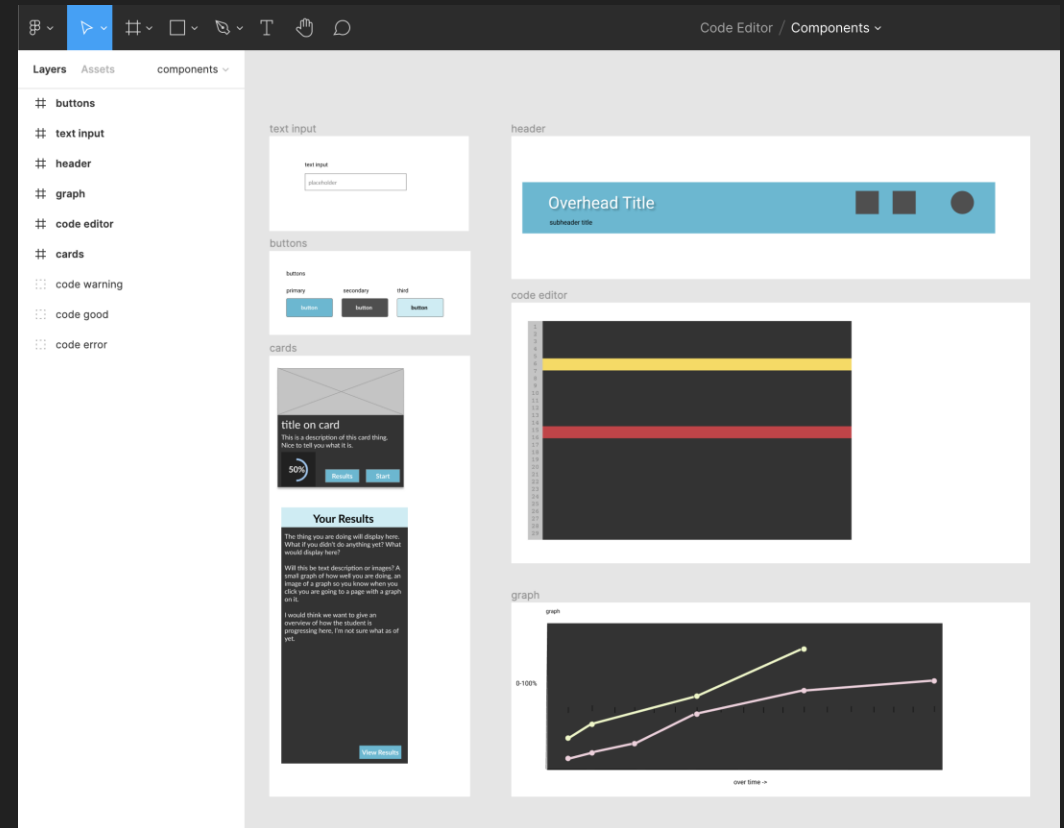
- Project Lead: Lisa Walkosz-Migliacio
- Front End team:
  - Roshni Dhanasekar, Khushi Magiawala, Vikas Barevadia

# Motivation

- Create a user interface with challenges that allow students to learn JavaScript in a code editor
- Provide useful feedback on compile and run-time errors in natural and easy-to-understand language
- Save and present user progress to provide a meaningful learning experience

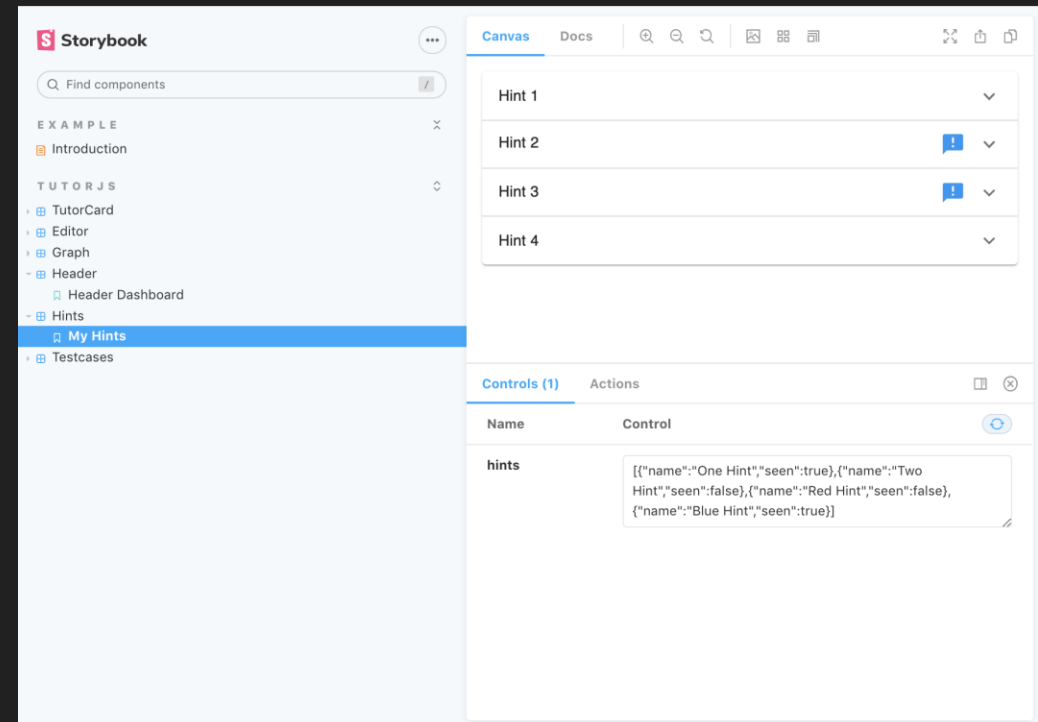
# Project Outline

- Research and Needfinding
- **Figma** - To Make Wireframes
- **GitHub Kanban board** – To organize action items for the code editor development



# Project Outline

- **React on Node.js** - for the Application
- **Material UI** - for Icons and standard Components
- **Storybook** - for the Components we created for the Application
- **Ace Editor** - for the Code Editor
- **D3** - for the Graphical Results
- **Axios** – to create a service to talk to the API



# Research Study Outline

**Mission:** To measure the success of our application based on the questions we used to guide its functionality

## Questions:

- How students best learn a coding language when much of the debugging is in exception errors?
- How students best learn a coding language when given multiple test cases for validation and practicing those failures again?
- How students best learn a mathematical formula from code and graphical visualization?

## Things To Test:

1. Effectiveness of Hints
2. Human Readable Error Messages and Stack Trace with Helpful Links
3. Overall UI (Ease of Use) (ex: layout of testcases next to code editor)

**Assessing Risk:** Emotional Risk (losing confidence in coding ability)

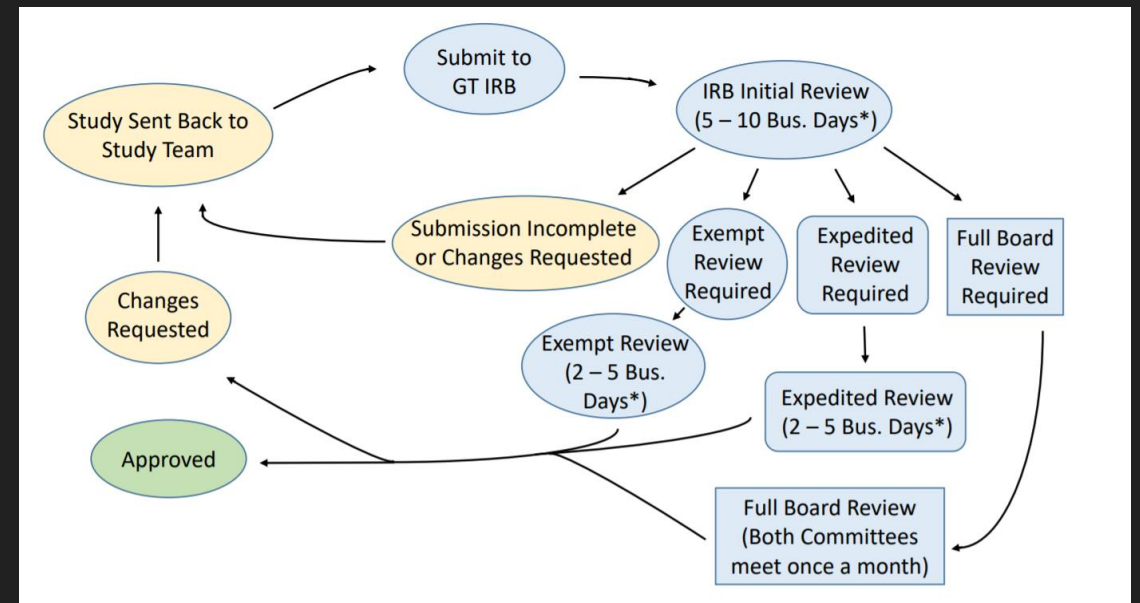
**Potential Subject Population:** Experienced vs Beginner Coders (establish a baseline), GT Students / ECE Students

# CITI and the Institutional Review Board

## CITI Relevant Concepts

- **Confidentiality:** remove identifiers from study (name, age, major, email) and substitute with participant IDs
- **Informed Consent:** appropriate language level
- **Potential benefits:** helpful testcases > get questions correct > feel more confident in coding ability
- **Minimize emotional risk:** inform about difficulty level and emphasis that this is a test run of a possible solution, not a proven solution
- **Waivers of Documentation:** Study participation presents **minimal risk of harm** to the subject and the research involves **no procedures requiring consent outside the context of participation**

## GT Central IRB Protocol



# Frontend: Gather Insights

- Gathered insights from various CS problem solving platforms (ex. Khan Academy, Leetcode)
- Analyzed these platforms and formed a list of feature to include in our application

## Challenge: Favorite Fruits


### Make the array

You're going to display your top 3 favorite fruits. To get started, make an array of them.

**Hint:** The fruit names are *strings*, so don't forget to write them between "quotation marks".

**Hint** [What's this?](#)

[Report a problem](#)

var  = [ ....., ....., ..... ];

```
1 var fruit = ["apple", "orange", "pear"];
2 |
```

Nice work!  
Great things are  
ahead!



Undo

Start over



Step 1/3

Next step



# Frontend: Features List

- Main Editor Screen
- Place for Hints
- Description Box + Examples
- Progress Bar
- Submit, Undo, Clear, Next Buttons
- Layout: Horizontal, Side-by-side

Mock Contest Discuss Store

Discuss (878) Submissions

JavaScript Autocomplete

```
1 + /**
2 +  * @param {number[]} arr
3 +  * @return {void} Do not return anything, modify arr in-place
4 +  */
5 + var duplicateZeros = function(arr) {
6 +     hi
7 + };
```

10 votes)

modified in-place. if in-place was not a constraint we might have just copied the elements from a source array to a

SOURCE ARRAY

1	0	2	3	0	4	5	0
---	---	---	---	---	---	---	---

FROM SOURCE TO DESTINATION

DESTINATION ARRAY

1	0	0	2	3	0	0	4	5	0	0
---	---	---	---	---	---	---	---	---	---	---

destination array is full.

]

that we do not grow the new array, rather we just trim it to its original array length. This means we have to discard by. These are the elements whose new indices are beyond the length of the original array.

SOURCE ARRAY

1	0	2	3	0	4	5	0
---	---	---	---	---	---	---	---

✕ Pick One < Prev 1089/1762 Next > Console Use Example Testcases

Your previous code was restored from your local storage. [Reset to default](#)

Testcase Run Code Result Debugger

Line 6 in solution.js  
hi  
~  
ReferenceError: hi is not defined  
Line 6: Char 5 in solution.js (duplicateZeros)  
Line 17: Char 19 in solution.js (Object.<anonymous>)  
Line 16: Char 8 in runner.js (Object.runner)  
Line 8: Char 26 in solution.js (Object.<anonymous>)  
Line 1208: Char 30 in loader.js (Module.\_compile)  
Line 1228: Char 18 in loader.js (Object.Module.\_extensions..js)  
Line 1049: Char 32 in loader.js (Module.load)  
Line 937: Char 14 in loader.js (Function.Module.\_load)  
at Function.executeUserEntryPoint [as runMain] (internal/modules\_line 17: Char 47 in run\_main\_module.js)

# Frontend: Wireframe

- Designed a wireframe for our all pages of our code editor application using Figma
- Took an example problem from Leetcode to outline our problem statement, hints, and test cases

**Problem Description:** Find the index of the maximum element in any given mountain array

*What is a mountain array?*

-> It is an array that is arranged in first ascending and then descending order

**Example 1:**  
Input: [0,1,0]  
Output: 1

**Example 2:**  
Input: [0, 2,1,0]  
Output: 1

**Example 3:**  
Input: [3,4,5,2]  
Output: 2

Test Case 1	★	
Test Case 2	★	
Test Case 3	✖	
Test Case 4	★	
Test Case 5	✖	🔒
Test Case 6	✖	🔒
Test Case 7	★	🔒
Test Case 8	✖	🔒
Test Case 9	✖	

```
1 class Solution {  
2   public int peakIndexInMountainArray(int[] arr) {  
3     index int = 0  
4   }  
5 }
```

Clear >

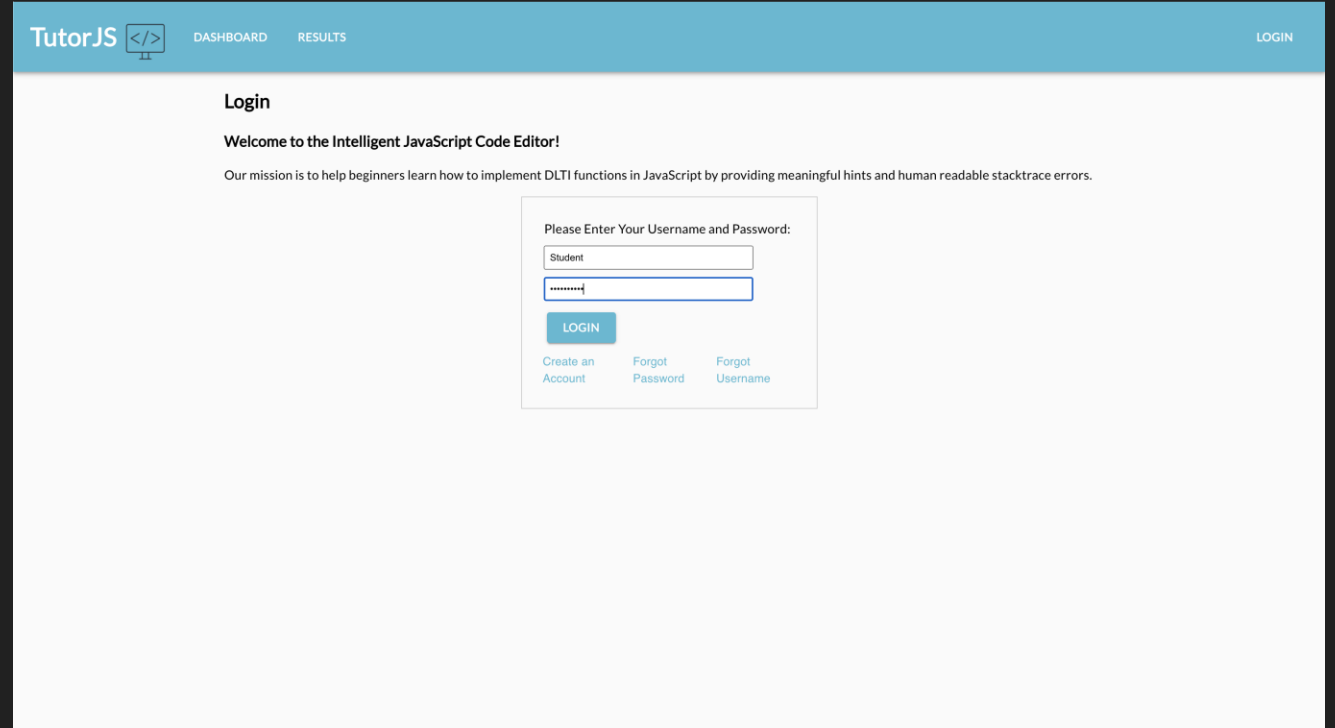
Undo >

Submit >

Next >

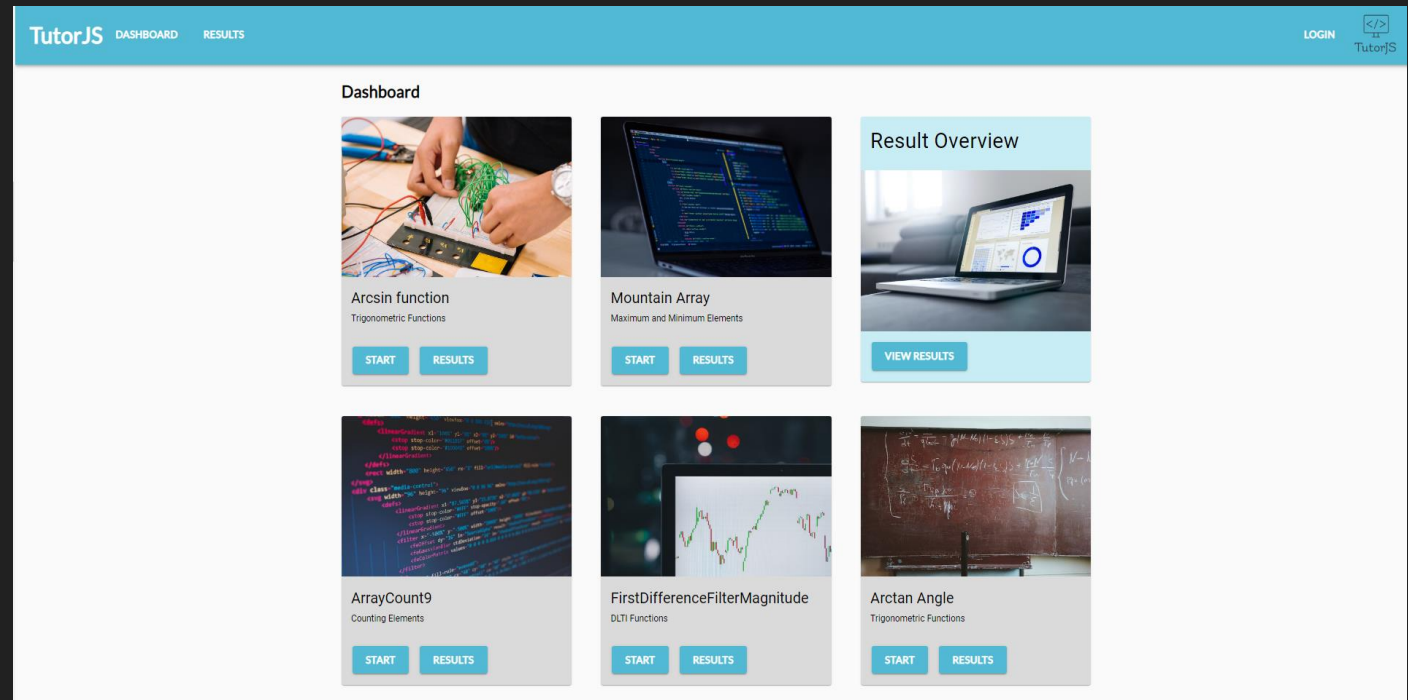
# Frontend: Login Page

- Welcomes user to the application with mission statement, login fields and account options
- A username and password can log a user into a session until they logged out keeping their work associated with their userid.



# Frontend: Dashboard Page

- Found other example problems on LeetCode and incorporated it into our code editor format
- Developed a Dashboard page that links to each challenge and completion results



# Frontend: Code Editor Page

- Created a functioning code editor that provides live feedback for syntax errors
- Learning hints, logging, and executed testcases results can be used when solving problems

The screenshot shows the TutorJS code editor interface. The top navigation bar includes 'TutorJS', 'DASHBOARD', 'RESULTS', and 'Welcome GBurdell2 LOGOUT'. The main content area is divided into three sections:

- Mountain Array:** The problem description asks to find the index of the maximum element in a mountain array. It provides three steps: 1. Declare a variable for the current index and set it to 0. 2. Structure a loop to cycle through the array until the maximum element is found. 3. Return the index of the maximum element. An example shows 'Input: arr = [0, 1, 0] Output: 1'.
- Hints:** A list of six hints is shown. Hint 1 is expanded, stating 'Variable should be an integer'.
- Code Editor:** A dark-themed editor with a light blue border. It contains the following code:

```
1 // DO NOT MODIFY THE FUNCTION HEADER!  
2 function mountainArray(arr) {  
3   // Write your code here  
4   // To print some value 'x', use 'log(x);'  
5  
6   log(arr);  
7   //make sure to return the correct value below  
8   return 0;  
9 }  
10  
11  
12  
13 mountainArray([0,1,0]);  
14 mountainArray([0,10,5,2]);  
15 mountainArray([24,69,100,99,79,78,67,36,26,19]);
```
- Output/Log:** A light gray box labeled 'log:' displays the output of the code execution:

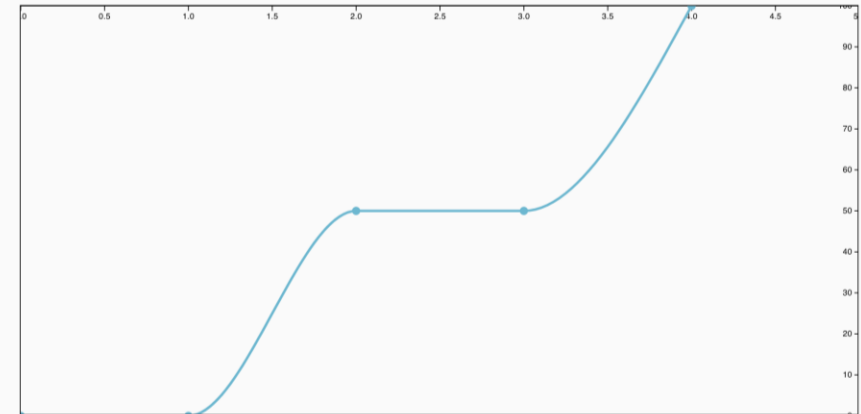
```
0,1,0  
0,10,5,2  
24,69,100,99,79,78,67,36,26,19  
0,1,0  
0,10,5,2  
24,69,100,99,79,78,67,36,26,19  
0,1,0  
0,1,0
```

At the bottom of the editor, there are buttons for 'RESET', 'UNDO', 'TEST', and 'SUBMIT'.

# Frontend: Results Page

- Graphical representation of submissions over time.
- Results report of the code, duration, hints, and test cases that passed or failed as well as error messages that were encountered.

Question [605e2506b4b6101224eaf708](#)



Results 2 Sat, 10 Apr 2021 14:35:30 GMT

## Statistics

duration: 12 minutes, 34 seconds

number of hints viewed: 3

percent test cases: 50%

## Code

```
// DO NOT MODIFY THE FUNCTION HEADER!
function arctanAngle(array) {
  if(array[1] === 5 && array[0] === 5) {
    return 45;
  }
  if(array[1] === 10 && array[0] === 10) {
    return 90;
  }
  return 0;
}

arctanAngle([5,5]);
arctanAngle([0,10]);
```

## Test Cases

Testcase 1:	✓
Testcase 2:	✗

# Future Features Frontend

- Currently using js-interpreter, which only runs vanilla JS
  - Look into other Ace Editor compilers, such as BabelJS
- Display all errors on respective lines in code editor
- Clean up and format the test cases

DEMO