

# MONKEYKRAFT CODERS

#### **Presentation 2023**

**Presented By:** 

Karishma Gawali Rithik Shetty Pranali Chipkar Mittul Sharma

CCCCCCC



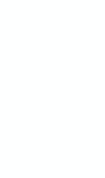
## Introduction

- Welcome to MonkeyKraft Coders, your all-encompassing platform dedicated to honing your coding skills.
- Our mission is to provide a rich variety of coding challenges that cater to all skill levels, ensuring a dynamic and effective learning experience.



#### Languages & Framework

- Typescript
- Javascript
- Css
- Python
- Html
- Mongoose
- MongoDB
- PostMan
- Express
- React
- BCrypt
- Acorn





#### Features

- Problem Sets
- Coding Playground
- Submission and Evaluation
- User Profiles





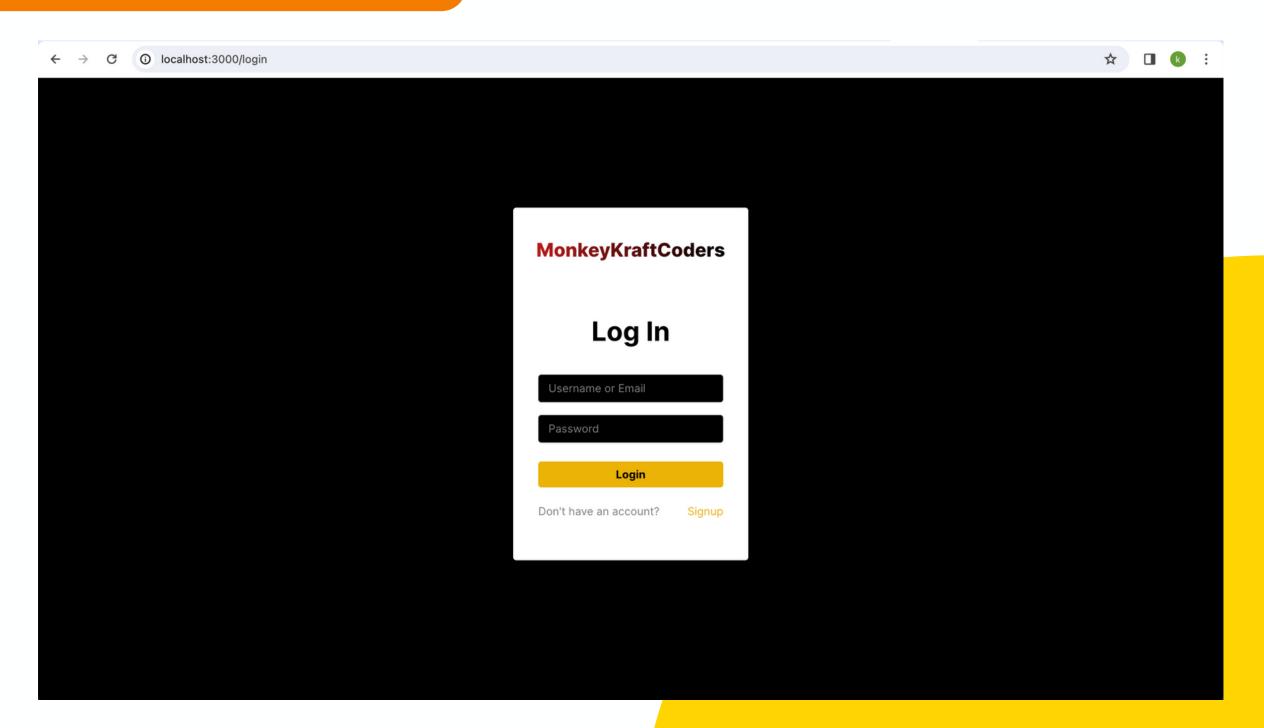
### Home Page

# **MonkeyKraft Coders** Sign Up Log In **Explore** Scale the heights of coding excellence alongside MonkeyKraft Coders. Advance your skills, triumph over challenges, and excel in technical interviews, propelling your programming journey to unparalleled success. **Get Started**





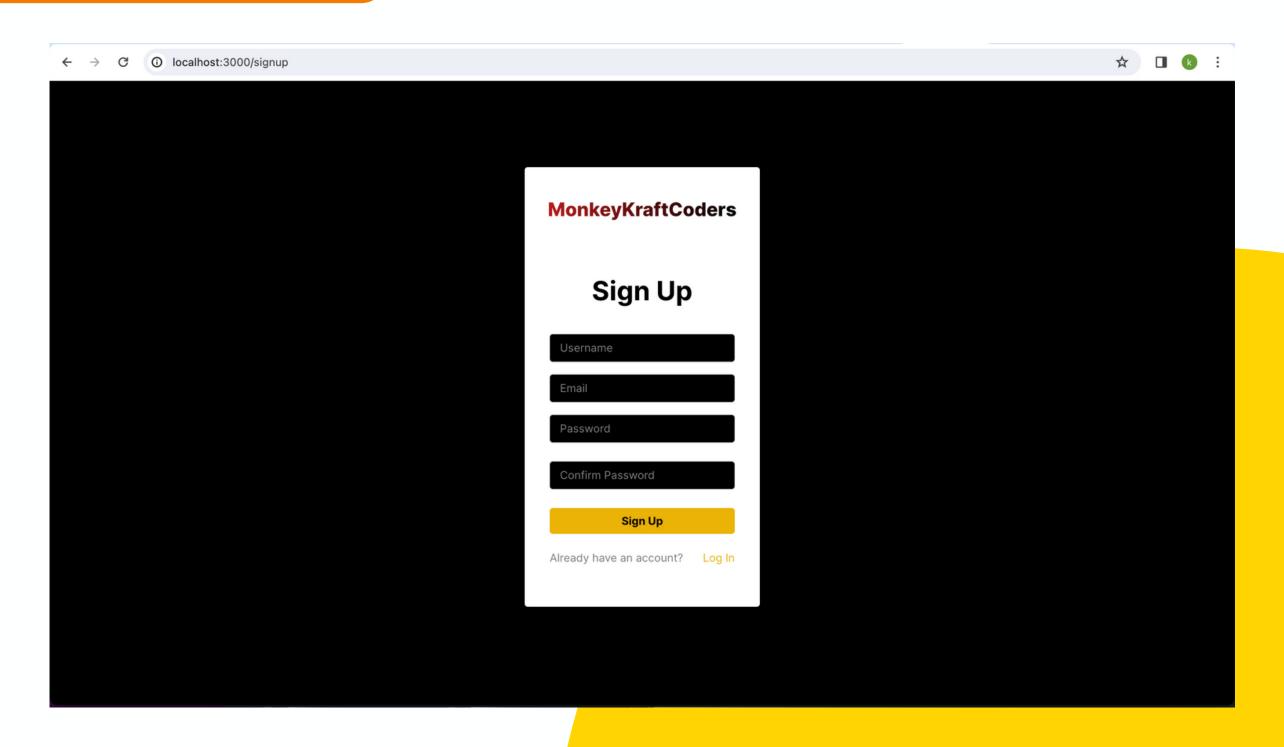
## Login Page







## Signup Page





## Problemset Page

#### MonkeyKraft Coders





All Topics Algorithms JavaScript DataBase Shell Search questions						
Status	Title♠	Acceptance 🔷	Difficulty <b></b>	Likes	Dislikes	Star
8	1. Maximum Subarray	92%	Easy	3	0	☆
$\otimes$	2. Minimum Subarray Sum	89%	Easy	7	1	☆
0	3. Longest Increasing Subarray	47%	Medium	4	0	☆
0	4. Subarray Sum Equals K	35%	Medium	11	0	☆
0	5. Shortest Subarray with Sum at Least K	5%	Hard	5	2	☆
0	6. Maximum Average Subarray I	71.4%	Easy	5	2	☆
0	7. Maximum Sum Circular Subarray	100%	Medium	3	0	☆
0	8. Longest Subarray with Ones after Replacement	62.5%	Medium	5	2	☆
0	9. Maximum Subarray Difference	50%	Hard	5	2	☆
0	10. Partition Array into Disjoint Intervals	50%	Medium	5	2	☆



### Solution Article

let minLength = Number.POSITIVE\_INFINITY;

for (let end = 0; end < nums.length; end++) {</pre>

minLength = Math.min(minLength, end - start + 1);

currentSum += nums[end];

while (currentSum >= k) {

let currentSum = 0;

let start = 0;

#### **MonkeyKraft Coders**

Problem List





Solution Article

Approach: Sliding Window

Algorithm

The sliding window technique is used to find the length of the shortest contiguous subarray with a sum at least k. It involves maintaining a window of elements and moving the window to find the shortest subarray.

The key idea is to keep track of the current sum of the window. If the sum becomes greater than or equal to k, we shrink the window from the start to find the shortest subarray with the required sum.

Implementation

var shortestSubarraySumAtLeastK = function(nums, k) {

```
javascript
6 var shortestSubarraySumAtLeastK = function(nums, k) {
                                                                              Submit
```



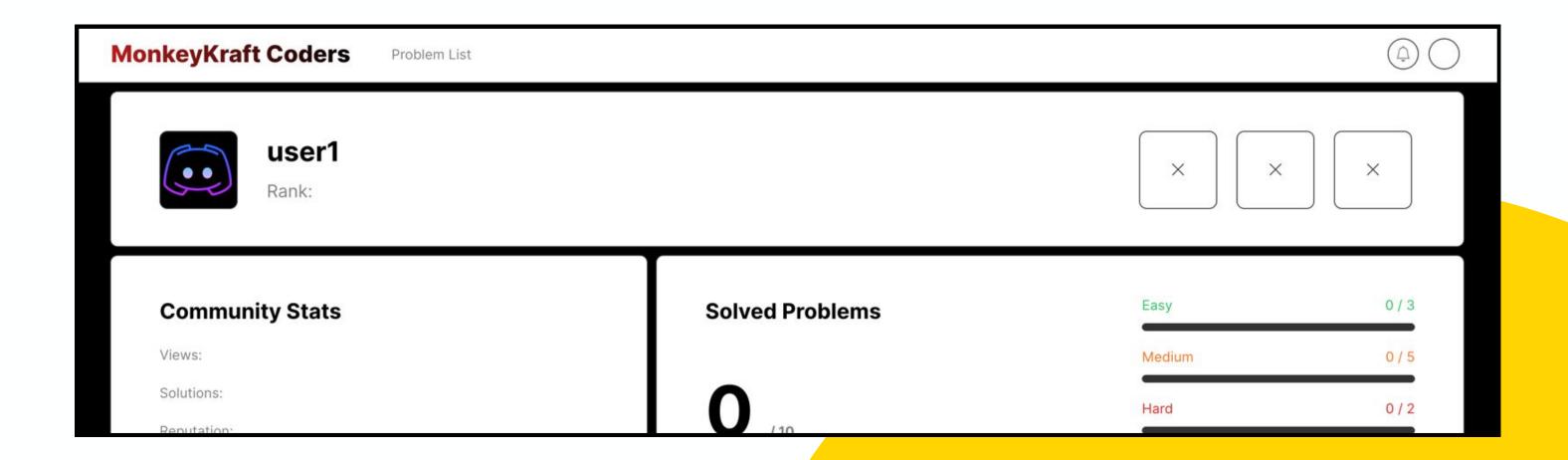
Submit

## Submission Page

#### **MonkeyKraft Coders** Problem List Description Editorial Submissions javascript Status Language Runtime Memory Date 2 \* @param {number[]} nums JavaScript 0ms 8MB December 12, 2023 5 var maxSubArray = function(nums) { JavaScript 0ms 50MB December 12, 2023



#### User





Thank You!