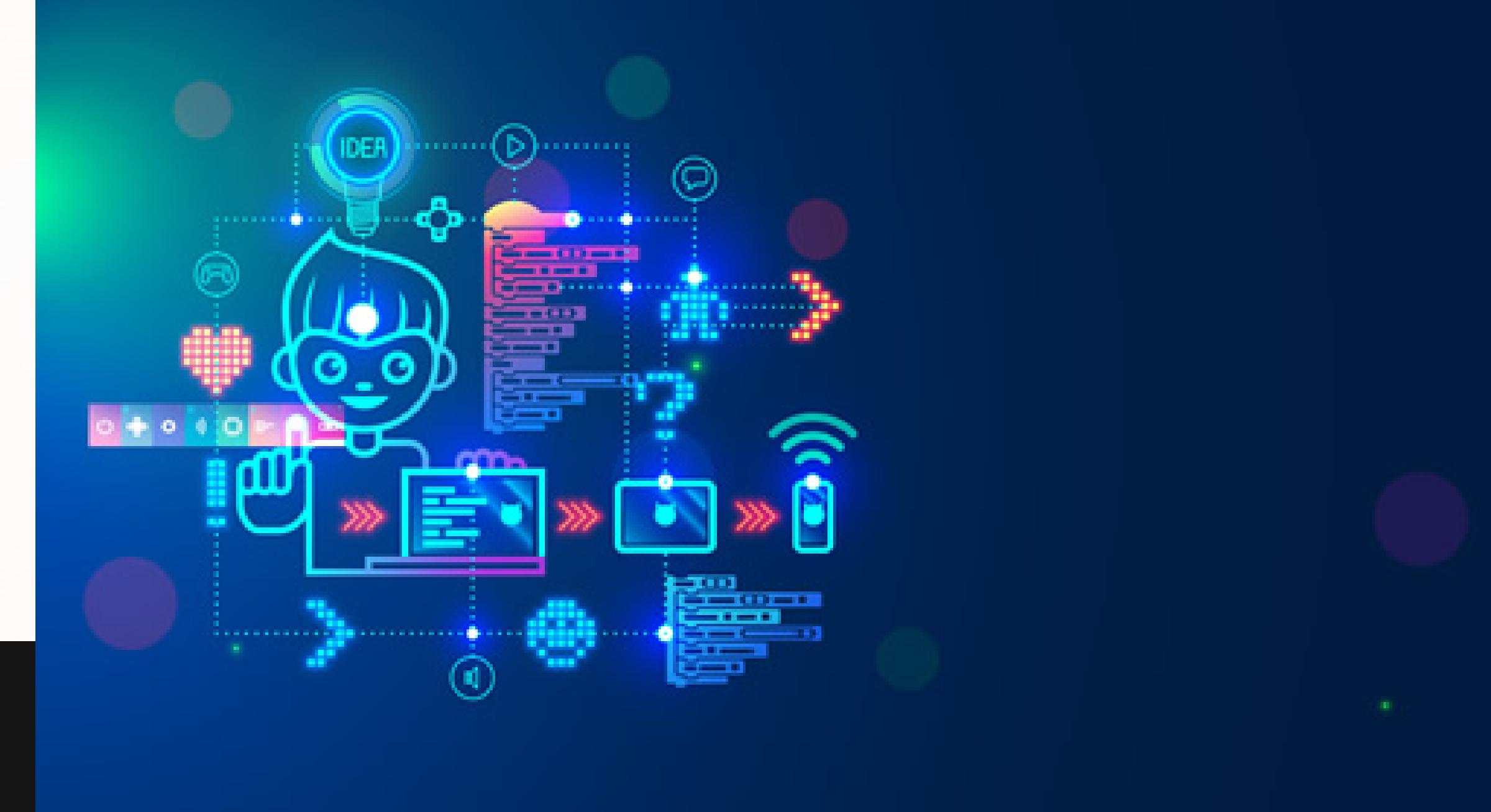


PRIYAMVADA
PRIYADARSHANI
IIB2020037

CODEBLOOM

CULTIVATING BRILLIANT
BLOOMS!

UNDER THE GUIDANCE OF:
PROF. NABAJYOTI MAZUMDAR



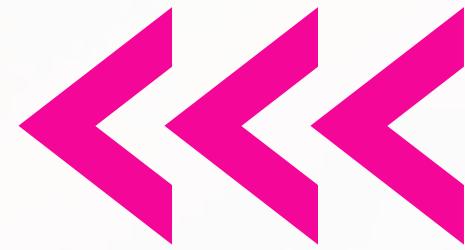
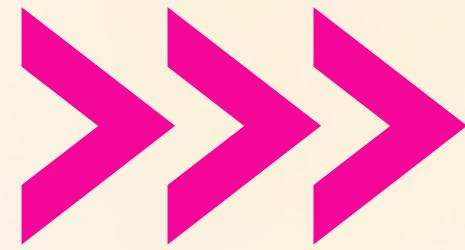


TABLE OF CONTENTS



01

INTRODUCTION

03

TOOLS &
TECHNOLOGY

05

FRONTEND
WALKTHROUGH

02

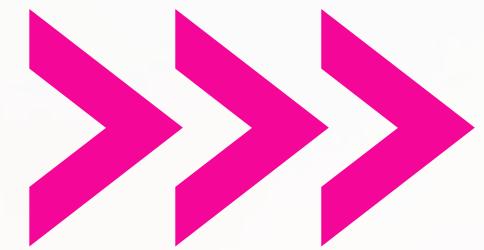
BACKEND
WALKTHROUGH

04

DATABASE
SCHEMA

06

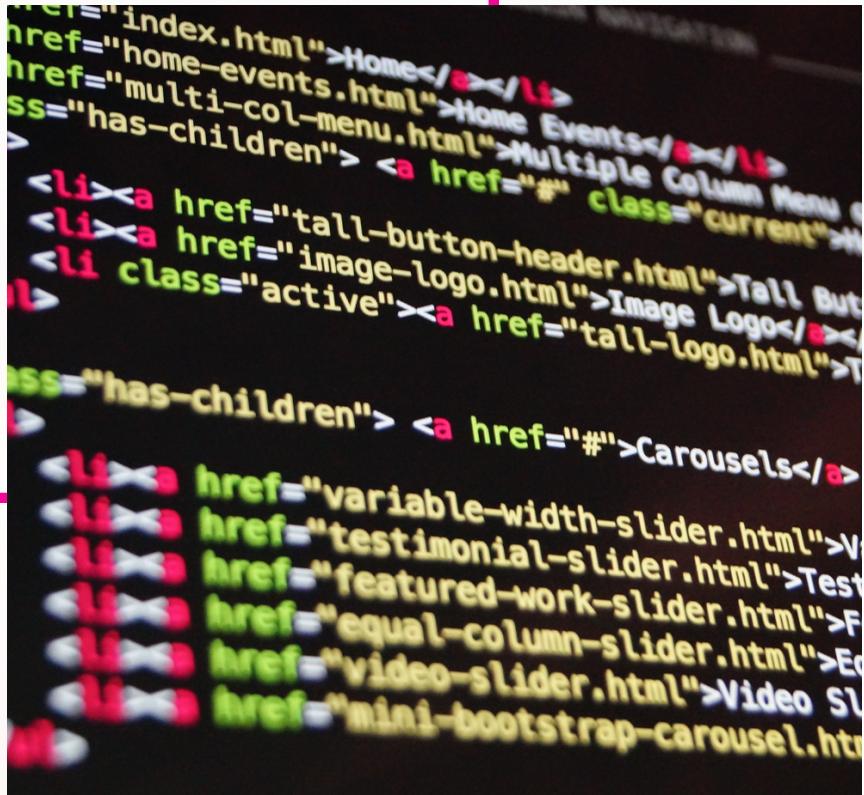
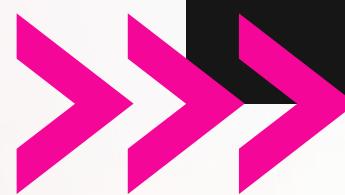
CONCLUSION &
LIVE LINK



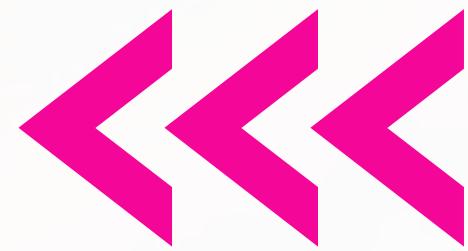
INTRODUCTION

CodeBloom is a full-stack coding platform where users can register and log in to submit their solutions and save their progress. It has various categories of questions like array, binary tree, etc., with difficulty levels such as easy, medium, and hard marked. Users may like, dislike, bookmark a problem, and view its video solution as well. Debugging is offered since errors are mostly displayed on a pop-up box. It combines static site generation and server site generation offered by next.js and makes use of some of the smoothest tools and technology in market to create a robust, efficient and user-friendly web application.

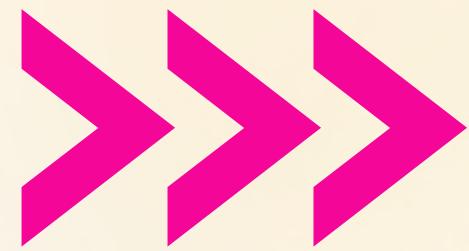
TOOLS & TECHNOLOGY



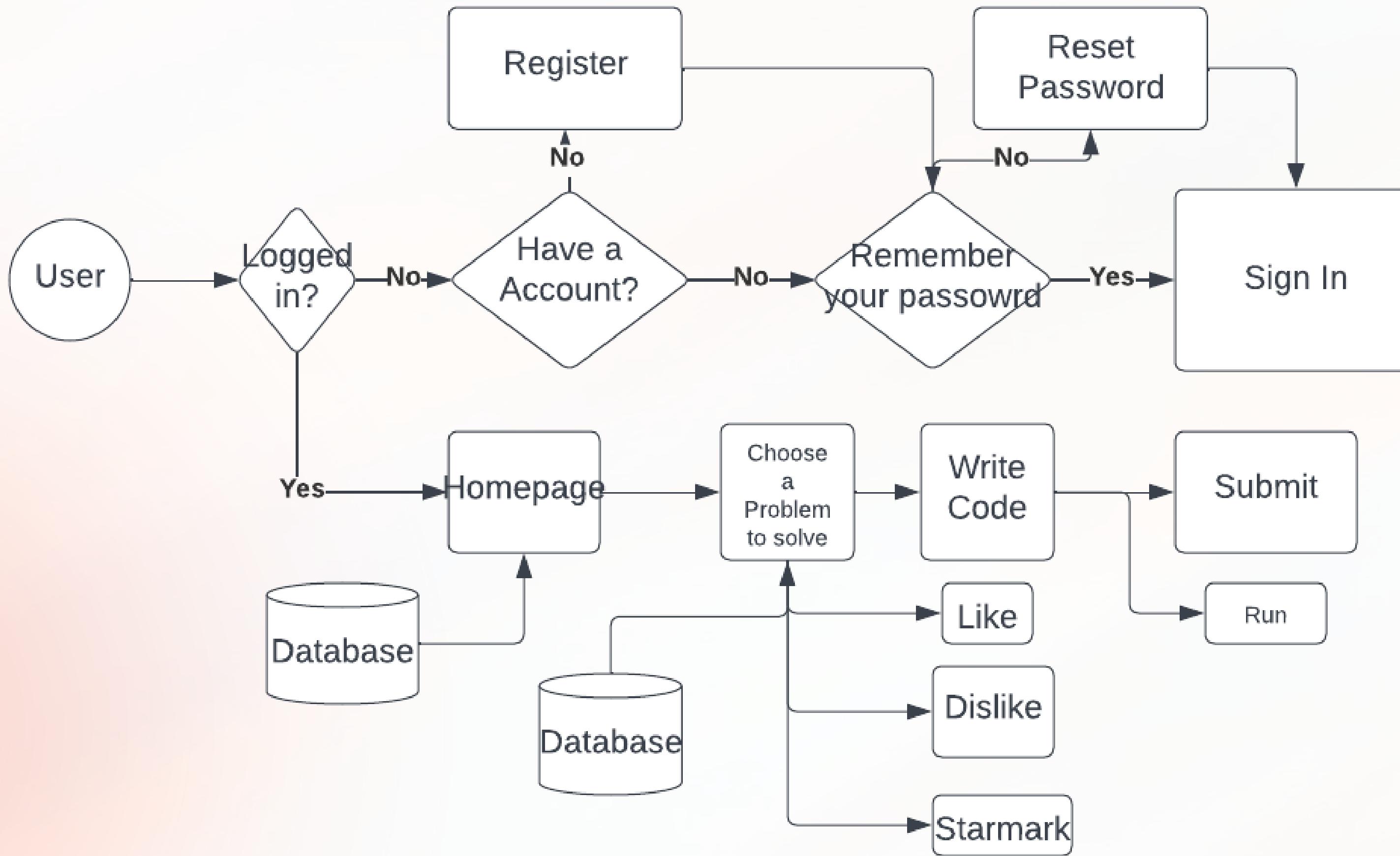
- 01 React
- 02 Next js
- 03 TypeScript
- 04 Tailwind css
- 05 Firebase
- 06 Vercel



FRONTEND WALKTHROUGH



Flowchart(Uml)



HOME PAGE

CodeBloom

Premium Sign In

"DEDICATION FUELS YOUR CODE, PRACTICE WITH PASSION AND WATCH IT EXPLODE!"

STATUS	TITLE	DIFFICULTY	CATEGORY	SOLUTION
	1.Two Sum	Easy	Array	
	2.Reverse Linked List	Hard	Linked List	Coming soon
	3. Jump Game	Medium	Dynamic Programming	Coming soon
	4. Valid Parentheses	Easy	Stack	Coming soon
	5. Search a 2D Matrix	Medium	Binary Search	Coming soon
	6. Container With Most Water	Medium	Two Pointers	Coming soon

Authentication

Sign in to CodeBloom

Your Email

Your Password

Log In

[Forgot Password?](#)

Not Registered? [Create account](#)

Register to CodeBloom

Email

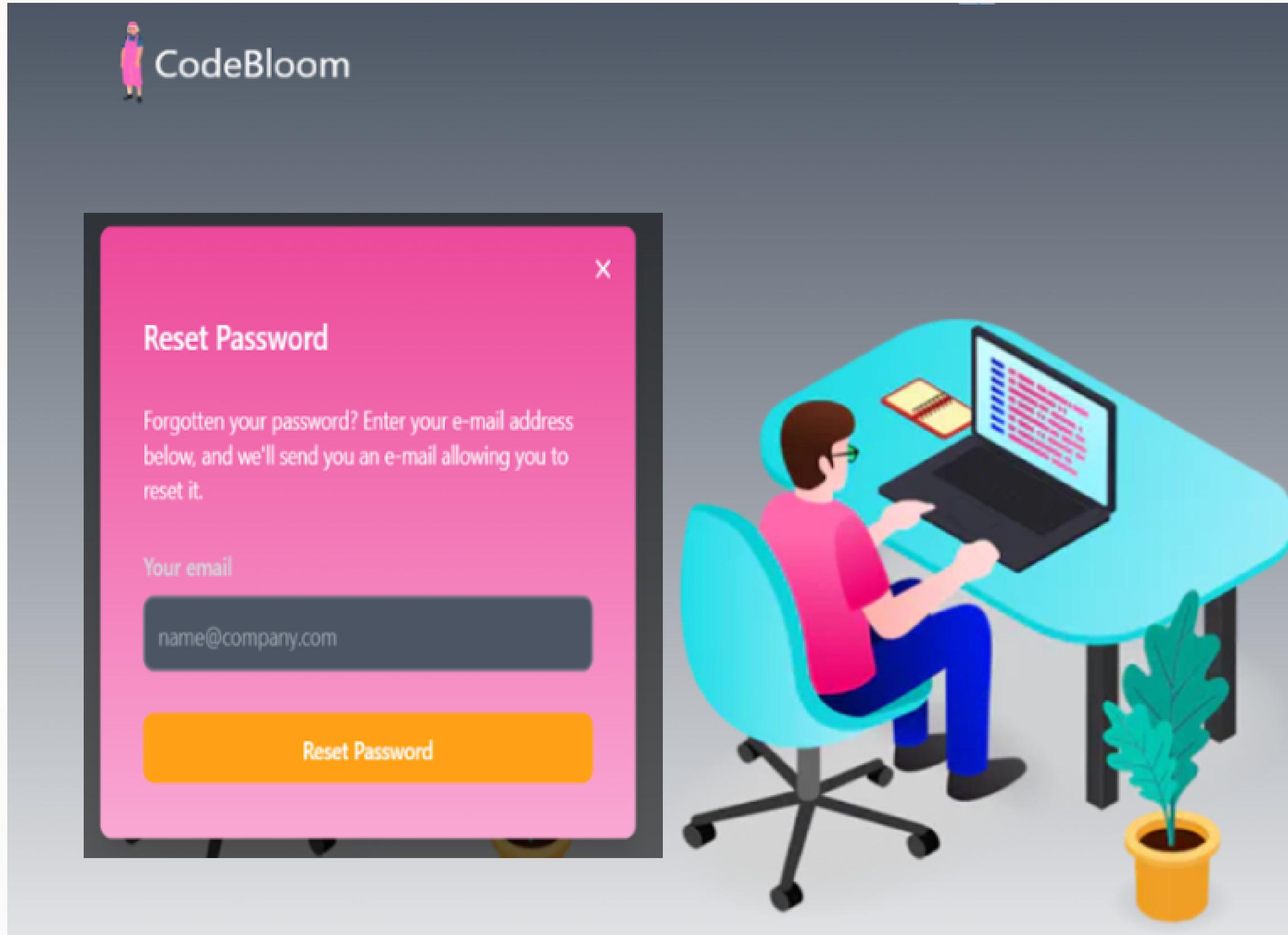
Display Name

Password

Register

Already have an account? [Log In](#)

Authentication



**Login/
ResetPassword/
Signup/
registration all
happen using
Firebase
auhtentication**

HOME PAGE



CodeBloom

Premium  

priyadarshanipriyamvada@gmail.com

"DEDICATION FUELS YOUR CODE, PRACTICE WITH PASSION AND WATCH IT EXPLODE!"

STATUS	TITLE	DIFFICULTY	CATEGORY	SOLUTION
✓	1.Two Sum	Easy	Array	
	2.Reverse Linked List	Hard	Linked List	Coming soon
	3. Jump Game	Medium	Dynamic Programming	Coming soon
✓	4. Valid Parentheses	Easy	Stack	Coming soon
	5. Search a 2D Matrix	Medium	Binary Search	Coming soon
	6. Container With Most Water	Medium	Two Pointers	Coming soon

Problem Page

CodeBloom

Problem List

Congrats! All tests passed!

Premium

Description

1. Two Sum

Easy ✓ 1 0

Given an array of integers `nums` and an integer `target`, return *indices of the two numbers such that they add up to target*.

You may assume that each input would have **exactly one solution**, and you may not use the same element twice.

You can return the answer in any order.

Example 1:

```
Input: nums = [2,7,11,15], target = 9
Output:[0,1]
Explanation: Because nums[0] + nums[1] == 9, we return [0, 1].
```

Example 2:

```
Input: nums = [3,2,4], target = 6
Output:[1,2]
Explanation: Because nums[1] + nums[2] == 6, we return [1, 2].
```

Example 3:

```
Input: nums = [3,3], target = 6
Output:[0,1]
```

Testcases

Case 1 Case 2 Case 3

Input:

```
nums = [2,7,11,15], target = 9
```

Console ^

Run Submit

AI

```
6 const num = nums[i];
7 const complement = target - num;
8
9 // If the complement is found in the map, return the indices
10 if (complement in complementMap) {
11     return [complementMap[complement], i];
12 }
13
14 // Store the current number and its index in the map
15 complementMap[num] = i;
16 }
17
18 // Return an empty array if no solution is found
19 return [];
20 }
21
```

Problem Page

CodeBloom

Problem List

ReferenceError: current is not defined

Description

2. Reverse Linked List

Hard ✓ 0 🔥 1 📈

Given the `head` of a singly linked list, reverse the list, and return *the reversed list*.

Example 1:

```
1  /**
2   * Definition for singly-linked list.
3   * function ListNode(val, next) {
4   *     this.val = (val===undefined ? 0 : val)
5   *     this.next = (next===undefined ? null : next)
6   * }
7   */
8   function reverseLinkedList(head) {
9     let prev = null;
10    // let current = head;
11
12    while (current !== null) {
13      const next = current.next; // Save the reference to the next node
14      current.next = prev;      // Reverse the next pointer
15
16      .....
17    }
18    return prev;
19  }
```

Summary

Testcases

Case 1 Case 2 Case 3

Input:

head = [1,2,3,4,5]

Output: [5,4,3,2,1]

Example 2:

Input: head = [1,2,3]

Output: [3,2,1]

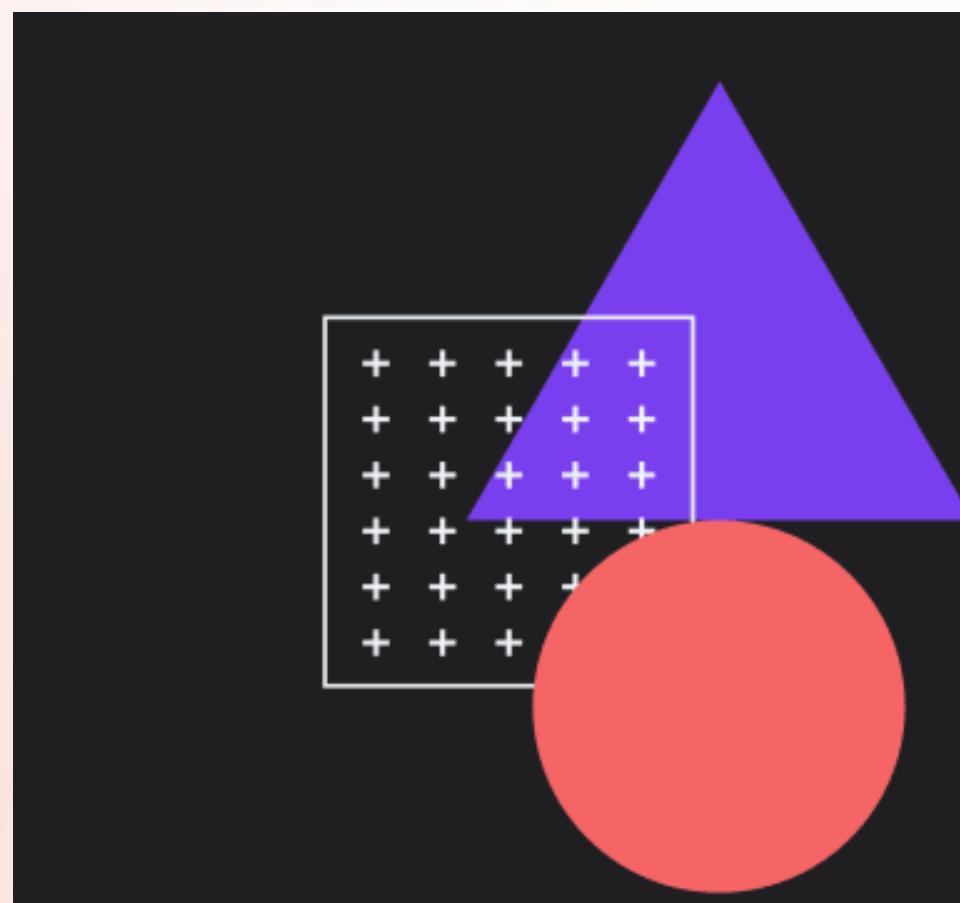
Console ▲

Run Submit

<< BACKEND >> WALKTHROUGH



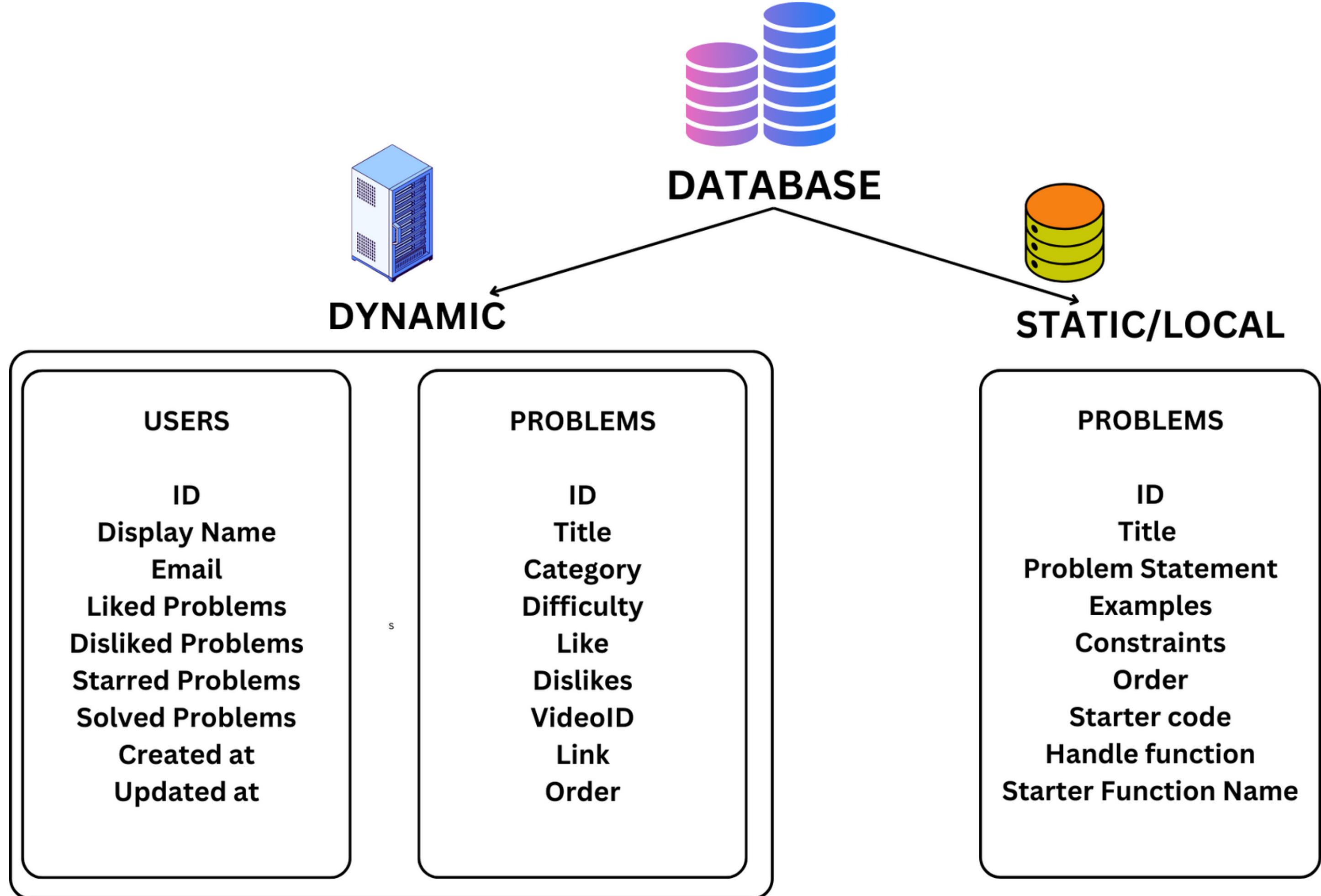
*Cultivate your code
with grace, witness it
flourish!*



web dev
**React +
TypeScript +
Tailwind CSS +
NextJS**

**How to decide
what data should
be locally present
on app server and
what should be
fetched from the
database server?**

**What are the
parameters of the
division of load? &
trade-off**



**Local data/
Problem
description
Eases page
loading and
reduces page
loading time.**

SCHEMAS:

```
export type Example = {
    id: number;
    inputText: string;
    outputText: string;
    explanation?: string;
    img?: string;
};

// local problem data
export type Problem = {
    id: string;
    title: string;
    problemStatement: string;
    examples: Example[];
    constraints: string;
    order: number;
    starterCode: string;
    handlerFunction: ((fn: any) => boolean) | string;
    starterFunctionName: string;
};

export type DBProblem = {
    id: string;
    title: string;
    category: string;
    difficulty: string;
    likes: number;
    dislikes: number;
    order: number;
    videoId?: string;
    link?: string;
};

export type Problem = {
    id: string;
    title: string;
    difficulty: string;
    category: string;
    order: number;
    videoId?: string;
};

export const problems: Problem[] = [
    {
        id: "two-sum",
        title: "Two Sum",
        difficulty: "Easy",
        category: "Array",
        order: 1,
        videoId: "8-k1C6ehKuw",
    },
    {
        id: "reverse-linked-list",
        title: "Reverse Linked List",
        difficulty: "Hard",
        category: "Linked List",
        order: 2,
        videoId: "",
    },
];
```

DATABASE SCHEMA

<p>+ Start collection</p> <p>problems ></p> <p>users</p>	<p>+ Add document</p> <p>container-with-most-water</p> <p>jump-game</p> <p>reverse-linked-list</p> <p>search-a-2d-matrix</p> <p>two-sum ></p> <p>valid-parentheses</p>	<p>+ Start collection</p> <p>+ Add field</p> <p>category: "Array"</p> <p>difficulty: "Easy"</p> <p>dislikes: 0</p> <p>id: "two-sum"</p> <p>likes: 1</p> <p>link: ""</p> <p>order: 1</p> <p>title: "1.Two Sum"</p> <p>videoId: "8-k1C6ehKuw"</p>
---	---	---

+ Start collection	+ Add document	+ Start collection
problems	r7sBq8dacoXhFtH4lIUfFd3EHEf1 >	+ Add field
users >		createdAt: 1687947929617

dislikedProblems

displayName: "Priyamvada2000"

email: "priyadarshanipriyamvada@gmail.com"

likedProblems

0 "two-sum"

solvedProblems

0 "two-sum"

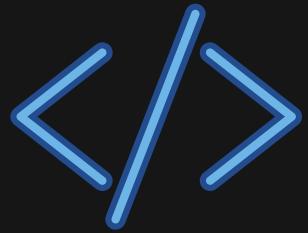
1 "valid-parentheses"

starredProblems

0 "two-sum"

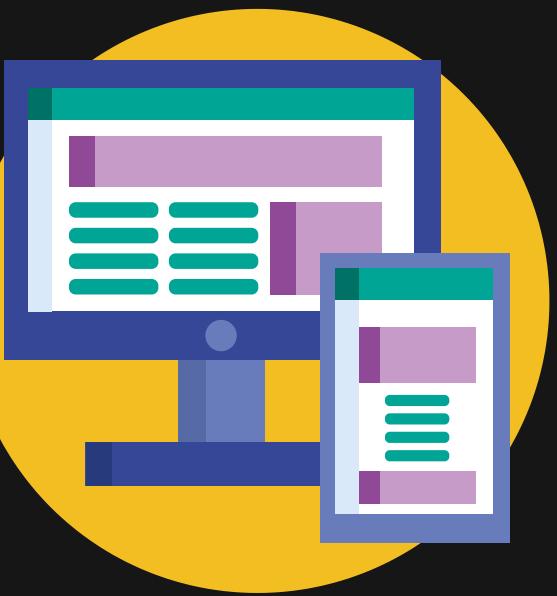
uid: "r7sBq8dacoXhFtH4lIUfFd3EHEf1"

updatedAt: 1687947929617

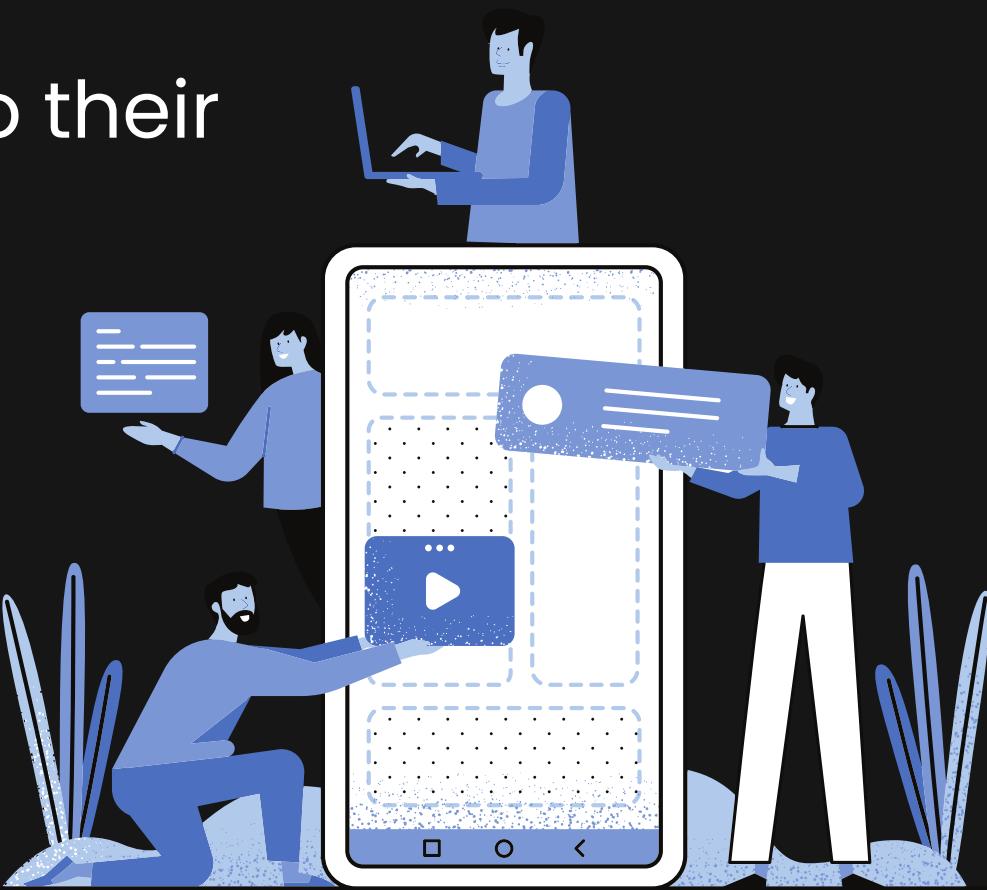


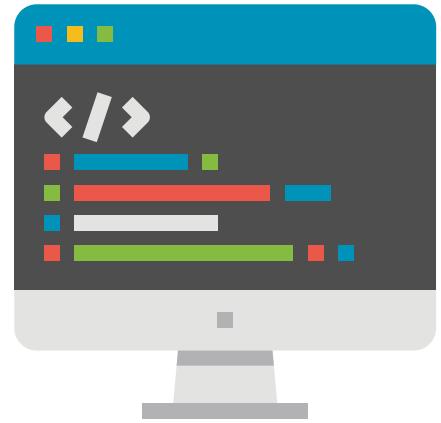
DEPLOYMENT

Deployed it on vercel.com to



- Ensure a seamless and efficient deployment process, as well as
 - To leverage its powerful features for hosting and
 - Scaling web applications.
-
- Vercel simplifies the deployment of web applications.
 - Its intuitive interface and automated deployment workflows
 - Enable developers to quickly deploy updates and changes to their applications
 - Without unnecessary complexity or manual configurations.





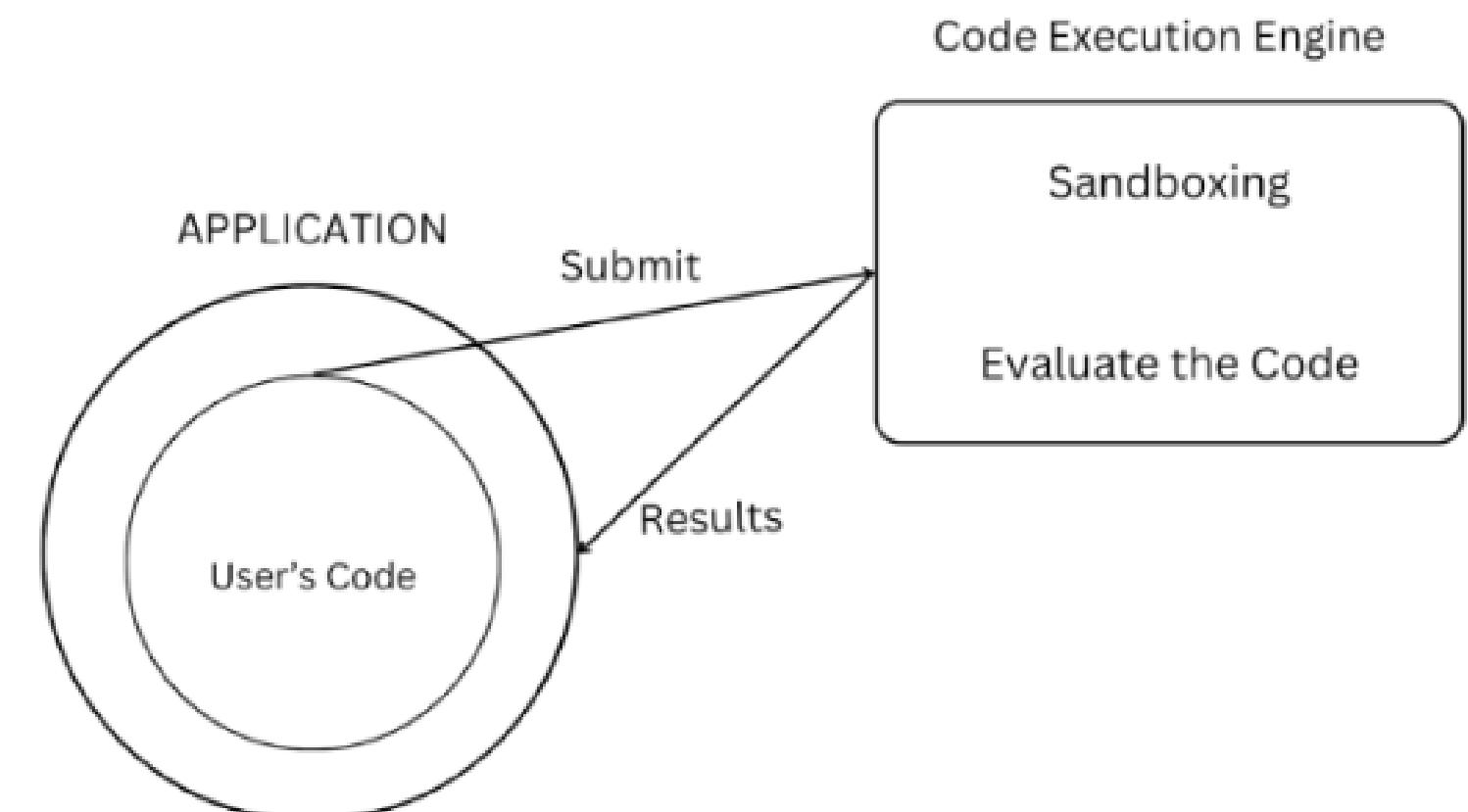
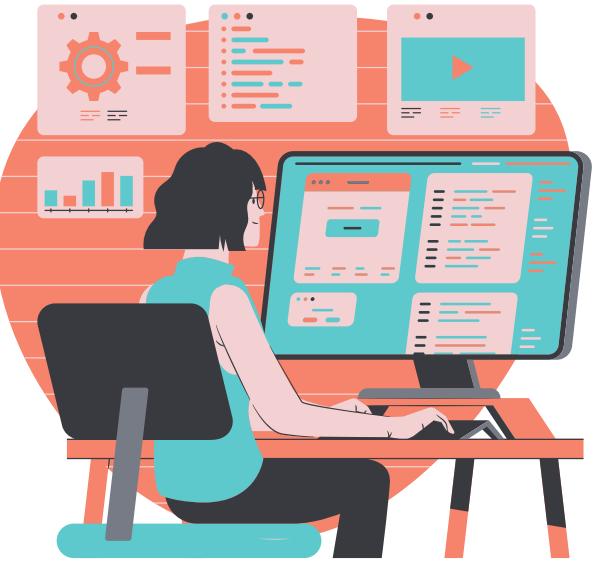
CONCLUSION

Achievements:

- Used transaction of Firebase to group multiple operations into a single transaction to maintain consistency.
- Made desired trade-offs using Dynamic and static storage options; integrating client-side rendering and SSG.
- Readable, Reusable, code.
- The diverse use of react packages to make user-friendly UI. CSS was quite challenging but tailwind CSS made things easier.
- Solved Hydration errors

Future Scope:

- Adding More Problems
- Adding More Section
- Expanding UI and adding more language support
- Sandboxing



LIVE LINK

<https://code-bloom.vercel.app/>

REFERENCES

- CodeBloom (code-bloom.vercel.app)
- Priyamvada-iiita/CodeBloom ([github.com](https://github.com/Priyamvada-iiita/CodeBloom))
- Firebase console ([google.com](https://console.firebase.google.com))
- react-toastify - npm ([npmjs.com](https://www.npmjs.com/package/react-toastify))
- react-confetti-explosion - npm ([npmjs.com](https://www.npmjs.com/package/react-confetti-explosion))
- code-bloom – Overview – Vercel
- React Icons ([react-icons.github.io](https://react-icons.github.io/react-icons))
- State management (<https://recoiljs.org/docs/>)
- <https://www.joshwcomeau.com/react/the-perils-of-rehydration/>



**THANK
YOU**