PROGRAMMING FOR KIDS

ARCHITECTURE AND API DOCUMENTATION

Group-03

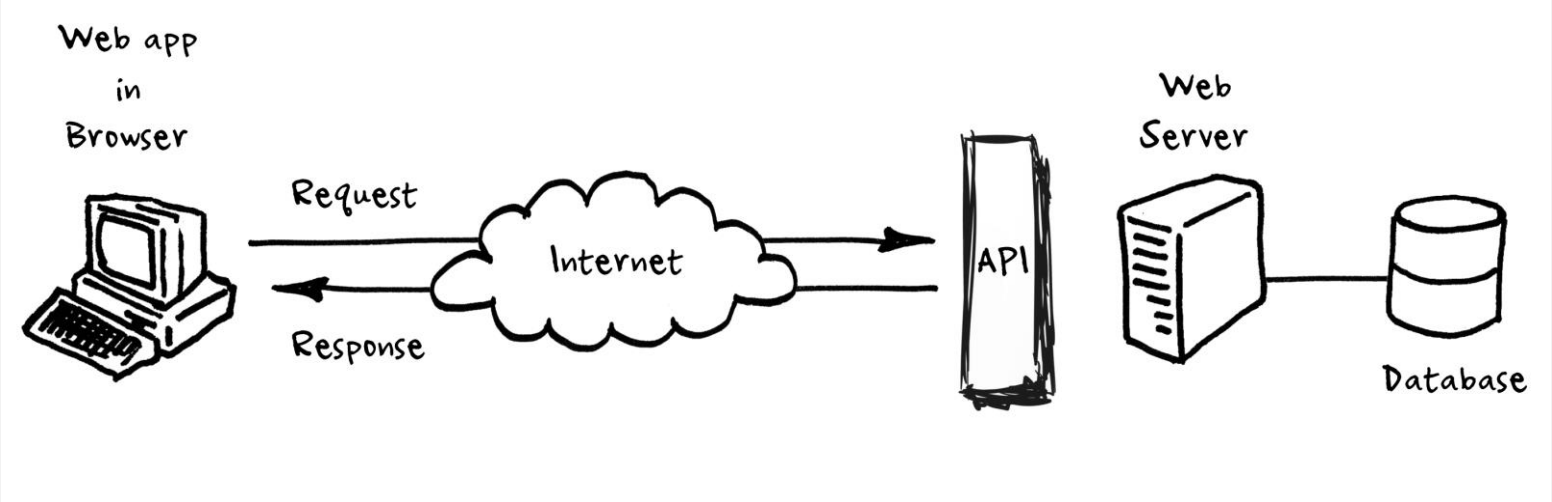
Submitted by-

1805095

1805096

1805118

ARCHITECTURE



TECH STACK

- **Frontend:**

- **Backend:**

- **Server:**

1.Getting user informations for DashBoard

API End-Point	HTTP Method	Request Body	Response Body

api/users/{userid
}
	GET	{
		
		}
			HTTP/1.1 200 OK
			{
			“username”:”Albert Einstein”,
			“name”:”Albert Einstein”,
			“rating”:23,
			“rank”: “Novice”
			registered_Contest:[{
			“contest_title”:”Array Round 1”
			“div”:3
			“contest_start_time”: “1d:05h:51m:38s”
			},
			{
			“contest_title”:”Linked List Round 2”
			“div”:3
			“contest_start_time”: “2d:05h:51m:38s”
			}]
			}

2. Updating Student Profile

API End-Point	HTTP Method	Request Body	Response Body

api/users/{userid}	PUT	{
		“id”:1
		“name”:”Albert Einstein”,
		“mobile_no”:”0123456789”,
		“email”: “einstein@gmail,com”,
		“address”: “Dhaka”
		“date_of_birth”: “1 Jan, 2000”
		“new_password”:”abcd6543”
		}
			HTTP/1.1 201 CREATED
			
			HTTP/1.1 204 NO CONTENT

3. Getting registered Contest information

API End-Point	HTTP Method	Request Body	Response Body

api/users/{user id}/registered-contests	GET	{
		
		}
			HTTP/1.1 200 OK
			
			{
			“registered_contests”:[{
			“contest_id”:1,
			“contest_title”:”Array Round 1”
			“div”:3
			“contest_start_time”:”1d:05h:51m”
			
			},
			{
			“contest_id”:2,
			“contest_title”:”Linked List Round 2”
			“div”:3
			“contest_start_time”:”2d:05h:51m”
			
			}]
			}

4. Getting Recommended Contest

API End-Point	HTTP Method	Request Body	Response Body

api/users/{user id}/recommend ed-contests	GET	{
		
		}
			HTTP/1.1 200 OK
			{
			“recommended_contests”:[{
			“contest_id”:4,
			“contest_title”:”Array Round 2”
			“div”:3
			},
			{
			“contest_id”:5,
			“contest_title”:”Linked List Round 3”
			“div”:2
			},
			{
			“contest_id”:6,
			“contest_title”:”Educational Round 2”
			“div”:1
			}]
			}

5.View Created Contests Informations (Admin):

API End-Point	HTTP Method	Request Body	Response Body

api/contests	GET	{
		
		}
			HTTP/1.1 200 OK
			{
			“created_contests”:[{
			“contest_id”:1,
			“contest_title”:”Array Round 2”
			“div”:3
			},
			{
			“contest_id”:2,
			“contest_title”:”Linked List Round 3”
			“div”:2
			},
			{
			“contest_id”:3,
			“contest_title”:”Educational Round 2”
			“div”:1
			}]
			}

6.Creating new Contests (Admin):

API End-Point	HTTP Method	Request Body	Response Body
api/contests	POST	{ “contest_title”:”Array Round 2” “Div”:3 “start_time”:”11:30am, 20-01-2023” “duration”:”120min” }	HTTP/1.1 201 Created

7.Modifying Contests (admin):

API End-Point	HTTP Method	Request Body	Response Body

api/contests/{contestID}	PUT	{
		“contest_title”:”Array Round 2”
		“Div”:3
		“start_time”:”11:30am,20-01-2023”
		“duration”:”120min”
		}
			HTTP/1.1 201 CREATED

8. Adding new problems into Contests (admin):

API End-Point	HTTP Method	Request Body	Response Body

api/contests/{contestID}/problems	POST	{
		“contest_id”:1,
		“name”:”Sasha and Array Coloring”,
		“problem_statement”: “If price of a toy is X and you paid taka Y to the shopkeeper, calculate how much money you will get back if you buy three of them.
		The first line of input is X and the second line is Y.Print the output”
		},
		“sample_input”: “10 50”,
		“sample_output”: “20”
		}
			HTTP/1.1 201 CREATED

9.Modifying problems of the Contests (admin):

API End-Point	HTTP Method	Request Body	Response Body

api/contests/{contestID}/problems/{problemID}	PUT	{
		“contest_id”:1,
		“name”:”Sasha and Array Coloring”,
		“problem_statement”: “If price of a toy is X and you paid taka Y to the shopkeeper, calculate how much money you will get back if you buy three of them.
		The first line of input is X and the second line is Y.Print the output”
		},
		“sample_input”: “10 50”,
		“sample_output”: “20”
		
		}
			HTTP/1.1 201 CREATED

11. View Individual Problem Description :

API End-Point	HTTP Method	Request Body	Response Body
api/contest/{contest_id}/problems/{problems_id}	GET	{ }	HTTP/1.1 200 OK { “contest_title”:”Array Round 1”, “div”:”3”, “problem_name”:”Shopping”, “description”:”If the price of a toy is X and you paid taka Y to the shopkeeper, calculate how much money you will get back if you buy three of them. The first line of input is X and the second line is Y. Print the output”, “Sample_input”: ”10 50” “sample_output”:”20” “start_time”:”12:40:31 PM 15-07-2023”

12. Submission :

API End-Point	HTTP Method	Request Body	Response Body

api/contest/{contest_id}/problems/{problem_id}/submission/{userID}	POST	{
		“userID”:1,
		“submittedCode”:”int main() { return 0;}”
		“language”:”C++”
		}
			HTTP/1.1 201 Created

13. Get Submission Status:

API End-Point	HTTP Method	Request Body	Response Body

api/contest/{contest_id}/problems/{problem_id}/submission/{userID}/status	GET	{
		}
			HTTP/1.1 200 OK
			{
			“userID”:1,
			“status”:”Accepted”
			}

14. Getting User Submission History:

API End-Point	HTTP Method	Request Body	Response Body

api/contest/{contest_id}/submission/{userID}	GET	{
		}
			HTTP/1.1 200 OK
			{
			“problem_list”:[
			“6”:{“2:15pm”,”Shopping”,”GNU C++”,”Accepted”},
			“5”:{“3:15pm”,”Apple tree”,”GNU C++”,”Accepted”},
			“4”:{“4:15pm”,”Sum array”,”GNU C++”,”Accepted”},
			“3”:{“2:15pm”,”Tracking Segment”,”GNU C++”,”Accepted”},
			“2”:{“5:15pm”,”OMSK Metro”,”GNU C++”,”Accepted”},
			“1”:{“7:15pm”,”Shopping”,”GNU C++”,”Wrong Answer”}]
			}

15. Getting all users Contest Standings:

API End-Point	HTTP Method	Request Body	Response Body

api/contest/{contest_id}/standing	GET	{
		}
			HTTP/1.1 200 OK
			{
			”standings”:[
			{“name”:“Sakib”,
			“Score”:[”10000”,”500”,”1000”,1500”,”2000”,”2500”,2500”]
			},
			{“name”:“Asif”,
			“Score”:[”9000”,”500”,”1000”,1500”,”2000”,”2500”,2500”]
			}
			]
			}

17. View mini project contest details:

API End-Point	HTTP Method	Request Body	Response Body

api/mini-projects/{miniProjectID}	GET	{
		}
			HTTP/1.1 200 OK
			{
			“title”:”Make A Calculator”,
			“project_details”:”1.You are given a skeleton code. Modify this code to make a calculator.
			2.Make sure that the button colors must be blue”
			“code”:”int main()
			{
			return 0;
			}”,
			“start_time”:”12.00pm,12-01-2023”
			}

18. Mini project contest Submit:

API End-Point	HTTP Method	Request Body	Response Body

api/mini-projects/{miniProjectID}/submission	POST	{
		“userID”:1,
		“submittedSolution”:”solution string”
		}
			HTTP/1.1 201 Created

19. Mini project contest Submission status:

API End-Point	HTTP Method	Request Body	Response Body

api/mini-projects/{miniProjectID}/submission/users/{userID}	GET	{
		}
			HTTP/1.1 200 OK
			{
			“userID”:1,
			“status”:”Accepted”
			}

21. Create a new Mini project contest (Admin):

API End-Point	HTTP Method	Request Body	Response Body

api/mini-projects	POST	{
		“title”:”Make A Calculator”,
		“project_details”:”1.You are given a skeleton code. Modify this code to make a calculator.
		2.Make sure that the button colors must be blue”
		“code”:”int main()
		{
		return 0;
		}”,
		“start_time”:”12.00pm, 12-01-2023”
		}
			HTTP/1.1 200 OK

21. Update the Mini project contest (Admin):

API End-Point	HTTP Method	Request Body	Response Body

api/mini-projects/{miniProjectID}	PUT	{
		“title”:”Make A Calculator”,
		“project_details”:”1.You are given a skeleton code. Modify this code to make a calculator.
		2.Make sure that the button colors must be blue”
		“code”:”int main()
		{
		return 0;
		}”,
		“start_time”:”12.00pm, 12-01-2023”
		}
			HTTP/1.1 200 OK

23. View Mini project contest standing :

API End-Point	HTTP Method	Request Body	Response Body

api/mini-projects/{miniProjectID}/standings	GET	{
		}
			HTTP/1.1 200 OK
			
			{
			“data”:[
			{
			“name”: “Asif”,
			“score”:95
			},
			{
			“name”: “Mahbub”,
			“score”:94
			}
			]
			}

24. Getting practice problems list:

API End-Point	HTTP Method	Request Body	Response Body

api/problems	GET		HTTP/1.1 200 OK
			{
			”problems”:[
			{“problemID”:1,
			“problem_name”:“Sasha and array coloring”,
			“difficulty”: “Easy”,
			“topic”: “Array”
			}
			]
			}

25. Getting practice problems Description:

API End-Point	HTTP Method	Request Body	Response Body

api/problems/{problemID}	GET	{
		}
			HTTP/1.1 200 OK
			{
			“problem_name”:”Shopping”,
			“description”:”If the price of a toy is X and you paid taka Y to the shopkeeper, calculate how much money you will get back if you buy three of them.
			The first line of input is X and the second line is Y.
			Print the output”,
			“status”:”Not Submitted”,
			“Sample_input”:”10 50”,
			“sample_output”:”20”
			}

25. Submit Practice problem:

API End-Point	HTTP Method	Request Body	Response Body

api/problems/{problemID}/submission	POST	{
		“userID”:1,
		“submittedSolution”:”
		int main(){return
		0;}”
		}
			HTTP/1.1 201 Created

25. Getting Practice problem Submission Status:

API End-Point	HTTP Method	Request Body	Response Body

api/problems/{problemID}/submission/users/{userID}/status	GET	{
		}
			HTTP/1.1 201 Created
			{
			“userID”:1,
			“status”:”Accepted”
			}

26. View Submitted Practice problems(History):

API End-Point	HTTP Method	Request Body	Response Body

api/users/{userID}/ practice-submissions	GET	{
		}
			HTTP/1.1 200 OK
			{
			data:[
			{
			“problemID”:1,
			“problemName”: ”name string”,
			“submittedTime ”:”1.18pm,1-5-2023”,
			“language”:”C+ +”,
			“status”:”Acce pted”
			},
			
			]
			}

27. Discussion:

API End-Point	HTTP Method	Request Body	Response Body

api/problmes/{problemId}/discussions	GET	{
		}
			HTTP/1.1 200 OK
			{
			"data" : [
			{
			"commentID":1,
			”parentCommentID”:0
			"userID":1,
			"comment":"some comment"
			}
			]
			}

28. Post a new Comment:

API End-Point	HTTP Method	Request Body	Response Body

api/problmes/{problemId}/discussions	POST	{
		"userID":1,
		"comment":"some comment",
		}
			HTTP 201 Created

29. Reply to a Comment:

API End-Point	HTTP Method	Request Body	Response Body

api/problmes/{problemId}/discussions/{commentID}	POST	{
		"parentCommentID":1,
		"userID":1,
		"comment":"some comment",
		}
			HTTP 201 Created

30. Delete a Comment (ADMIN)

API End-Point	HTTP Method	Request Body	Response Body
api/problmes/{problemId}/discussions/{commentID}	DELETE	{ }	HTTP 200 OK

31. View solution:

API End-Point	HTTP Method	Request Body	Response Body

api/problems/{problemID}/solutions	GET	{
		}
			{
			"solution":"solution string",
			"videoLink":"https://youtube.com/hu1eCgs",
			"relatedLink":[
			{
			"desc":how to add two integers",
			"link":"https://geeksforgeeks.com"}
			]
			}

32. Create a new Solution (Admin):

API End-Point	HTTP Method	Request Body	Response Body

api/problems/{problemID}/solutions	POST	{
		"solution":"solution string",
		"videoLink":"https://youtube.com/hu1eCgs",
		"relatedLink":[{"desc":"how to add two integers",
		"link":"https://geeksforgeeks.com"}
		]
		}
			HTTP 201 Created

33. Update a solution(Admin)

API End-Point	HTTP Method	Request Body	Response Body

api/problems/{problemID}/solutions	PUT	{
		"solution":"solution string",
		"videoLink":"https://youtube.com/hu1eCgs",
		"relatedLink":[{"desc":"how to add two integers",
		"link":"https://geeksforgeeks.com"}
		]
		}
			HTTP/1.1 200 OK

34. Daily Coding Puzzle View:

API End-Point	HTTP Method	Request Body	Response Body

api/daily-coding-puzzle/{date}	GET	{
		}
			HTTP/1.1 200 OK
			
			{
			"Puzzle_id":1,
			"puzzleDate":"1-05-2023",
			"puzzle_question":"Find the wrong with the code snippets?
			provide your solution.",
			"puzzle_code":"#include <stdio.h>
			int main()
			{
			printf("Programming for kids !")
			return 0;
			}"
			}

35. Daily Coding Puzzle (Admin):

API End-Point	HTTP Method	Request Body	Response Body

api/daily-coding-puzzle/{date}	POST	{
		"puzzle_question":"Find the wrong with the code snippets? provide your solution.",
		"puzzle_code":"#include <stdio.h>
		int main()
		{
		printf("Programming for kids !")
		return 0;
		}",
		“solution”:”soution string”
		}
			HTTP/1.1 201 Created

36. Submit Solution of Daily Coding Puzzle :

API End-Point	HTTP Method	Request Body	Response Body

api/daily-coding-puzzle/{puzzleID}/submission	POST	{
		“userID”:1,
		“submittedSolution”:”soution string”
		}
			HTTP/1.1 201 Created

37. Get Submission Status of Daily Coding Puzzle :

API End-Point	HTTP Method	Request Body	Response Body

api/daily-coding-puzzle/{puzzleID}/submission/{userID}/status	GET	{
		}
			HTTP/1.1 200 OK
			{
			“userID”:1,
			“status”:”Accepted”
			}

38. Code Gaming View:

API End-Point	HTTP Method	Request Body	Response Body

api/code-gaming/{problemID}	GET	{
		}
			HTTP/1.1 200 OK
			{
			“problemID”:”1”
			“problemName”:”Star Wars”
			“problemStatement”:”Let’s tell your program to shoot the closest alien. Copy/Paste the following code in the right place in the code editor:
			int main() {
			if(a>b) printf(“a”);
			else printf(“b”);
			}”,
			“startingCode”:”int main()
			{int a,b,c;
			while(1){ }
			return 0;
			}”
			}

39. Submission:

API End-Point	HTTP Method	Request Body	Response Body

api/code-gaming/{problemID}/submission	POST	{
		“userID”:1,
		“submittedCode”: ”int main()
		{int a,b,c;
		while(1){ }
		return 0;
		}”
		
		}
			HTTP/1.1 201 Created

40. Hint from AI from practice:

API End-Point	HTTP Method	Request Body	Response Body

api/problems/{problemID}/hints	POST	{
		“userID”:1,
		“prompt”:[{“How to use printf?”},
		{“How to use a variable?”}
		]
		}
			HTTP/1.1 201 Created
			
			{
			“answer”:””
			}

THANK YOU