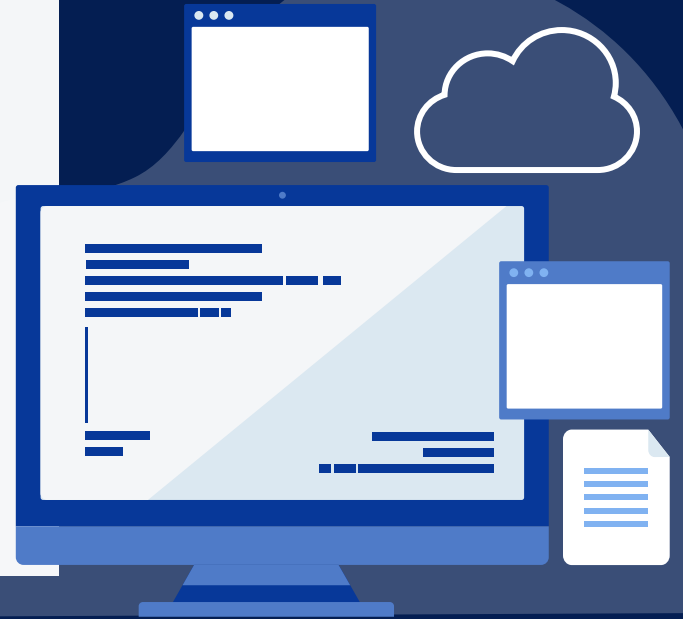




An Najah Rank

Software Graduation Project



Our Team



Supervisor
Dr. Samer Arandi



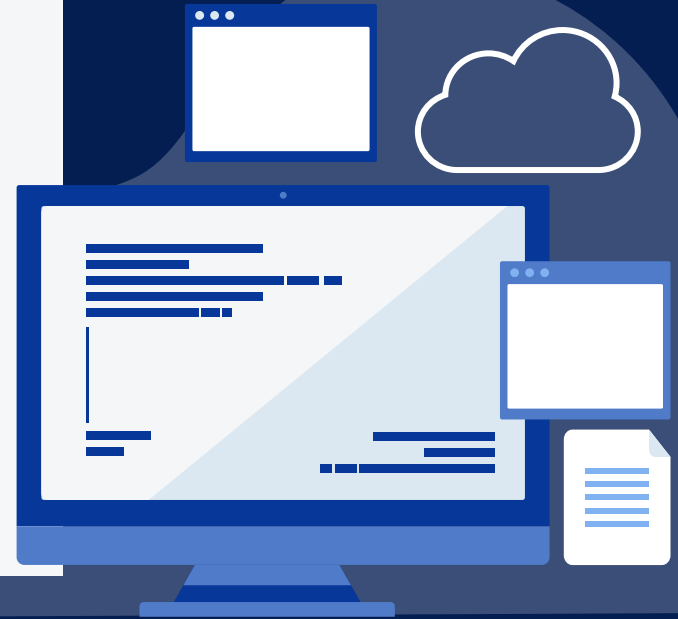
Momen Odeh



Noor Aldeen
Abu Shehadeh

Table of contents

- 01 Problem Statement
- 02 Our Solution
- 03 Features
- 04 Methodology
- 05 Challenges and Constraints
- 06 Future Work



What 's the problem?

The problem-solving skills are one of the most important skills in the workplace.

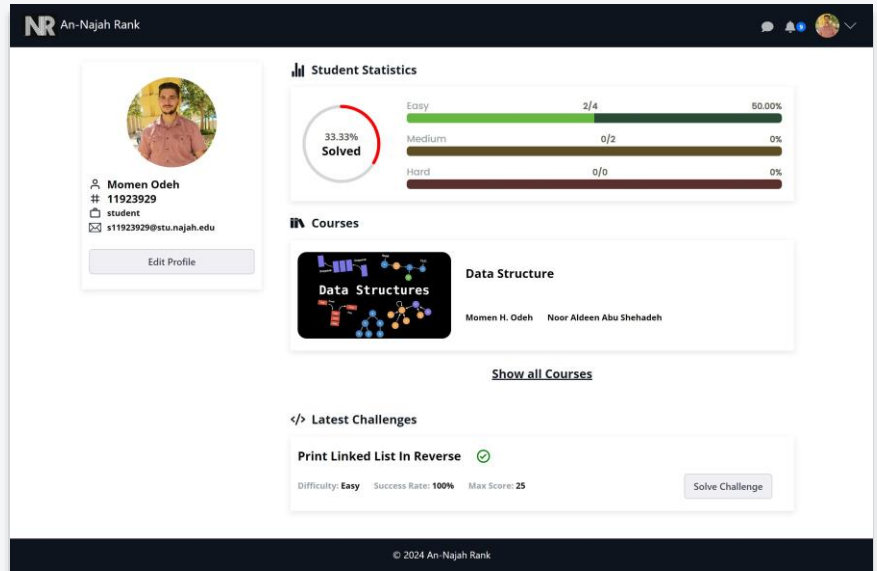
An-Najah University strives to improve these skills in our students.

Problem solving websites lack essential features that would simplify the problem-solving process and make solution grading more efficient.



Our Solution

We built An Najah Rank, a problem-solving web application that combines solving problems for students and adds the educational features needed for professors, making the process more simple.



Who use web application ?



Professors



Students



Admin

Features

01

Registration & login

02

Admin

03

Course Management

04

Manage test cases
of Challenge

05

Code operation

06

Similarity

07

Track submissions

08

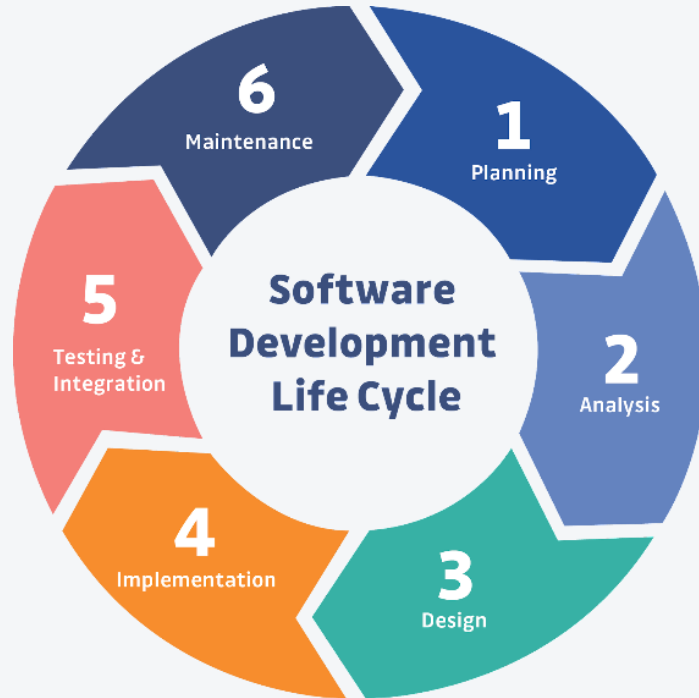
Manual Mark

09

Chatting & Notification

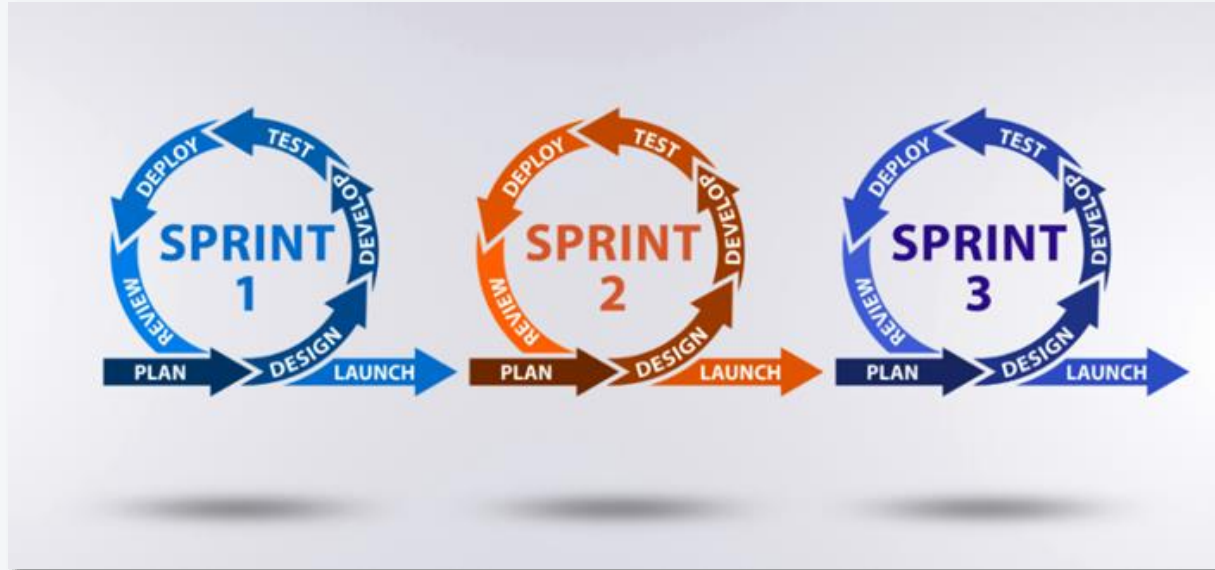
Methodology

Software Development Life Cycle:



Methodology (cont.)

Agile Methodology:



Planning Phase

- ❑ We met with our supervisor Dr. Samer Arandi.
- ❑ We explored various problem-solving websites.
- ❑ We are discussing new functionalities to add to the project.



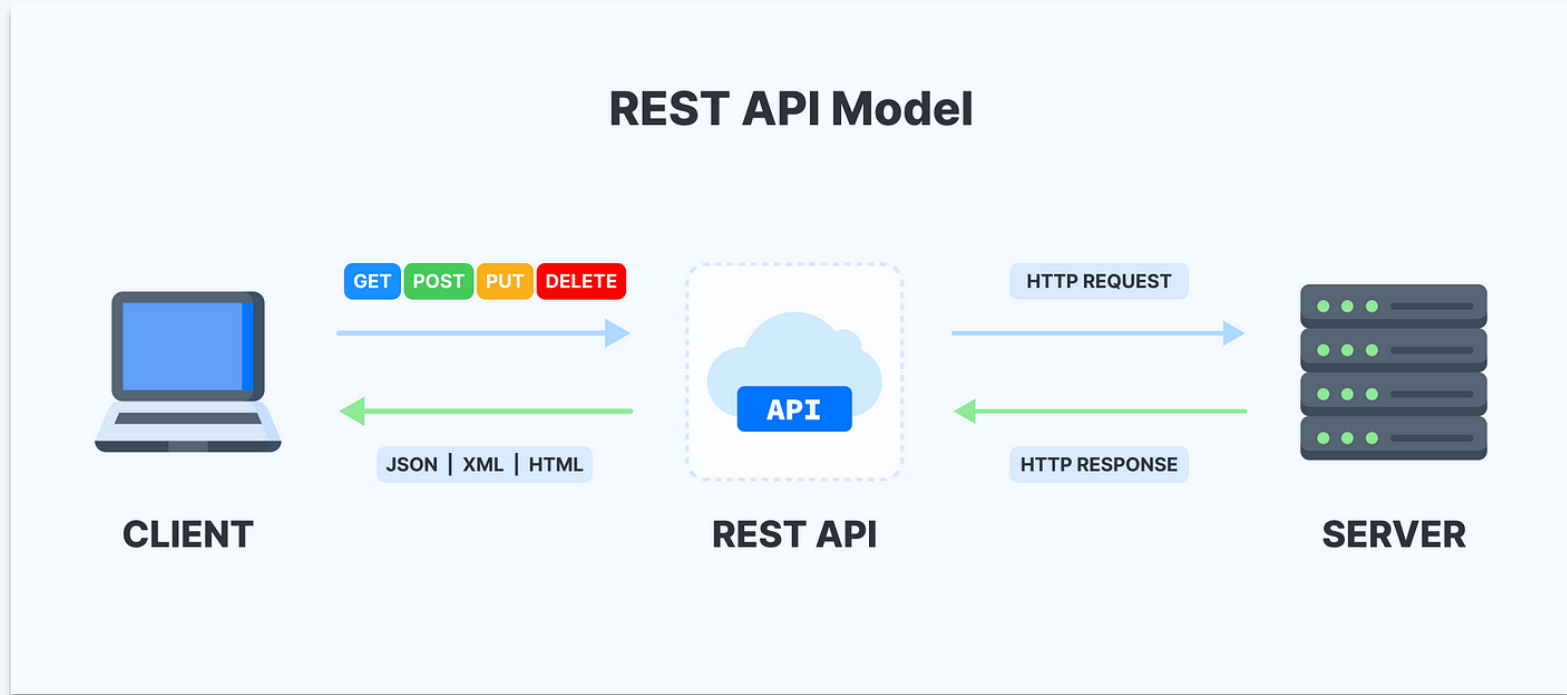
Analysis Phase

- ❑ Gathering and documenting requirements.
- ❑ Writing user stories to clearly outline specific functionalities.
- ❑ Envisioning the system's architecture using Unified Modeling Language (UML) diagrams.



Design Phase

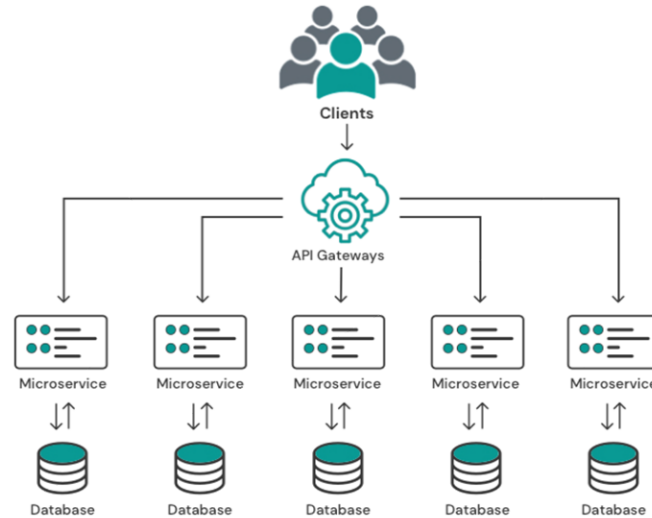
Architectural Style:



Design Phase (cont.)

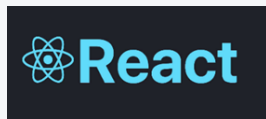
Architectural Pattern:

Microservice Architectural pattern



Design Phase (cont.)

Frontend libraries:



Backend technologies :



Design Phase (cont.)

DevOps tools:

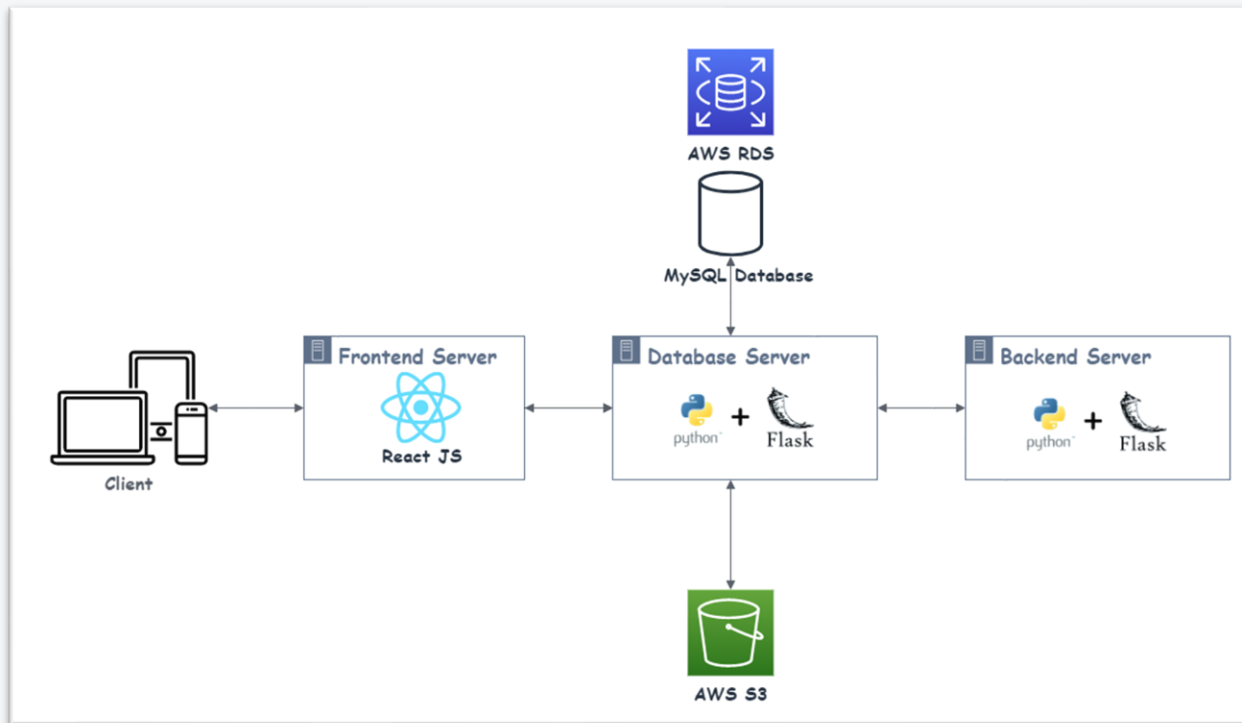


AWS Cloud Formation:



Design Phase (cont.)

Project Structure:



Security

- a) **Authentication:** verifying a user or device before allowing access to web application.
- b) **Authorization:** giving the user permission to access a specific page.
- c) **CORS policies:** a mechanism that allows restricted resources on a web page to be accessed from another domain outside the domain from which the first resource



Library used:



User Features

Registration:

NR An-Najah Rank

Sign Up

If you already have an account register
You can [Login here](#) !

Email

☐ Enter your email address


Full Name

Enter your Full Name


University Number

Enter your University Number

Password


Enter your Password 

Confirm Password

Confirm your Password 

☐ Sign up as professor

Register



Sign Up to An-Najah Rank

make your future

© 2024 An-Najah Rank

User Features

Sign in:

NR An-Najah Rank



Sign in to An-Najah Rank
make your future

Sign in

If you don't have an account register
You can Register here !

Email

✉ Enter your email address

Password

🔒 Enter your Password



[Forgot Password ?](#)

Login

User Features

Forget password:

NR An-Najah Rank

Forget Password

No Problem! Enter your email below and we will send you an Code with instruction to reset your password.

Reset Password

Back to Login

NR An-Najah Rank

New Password

Set the new password for your account so you can login and access all feautres.

New Password

Confirm Password

UPDATE PASSWORD


User Features

Account Settings:

Account

Password

General Info



st1923929@stu.najah.edu

11923929

student

Uplode image

Delete Image

Full Name

Momen Odeh

Delete Accounts

Delete your account and all information related to your account such as your profile page, badges earned and leaderboard positions. Please be aware that all data will be permanently lost if you delete your account.

Delete Account

✓ updated successfully

×

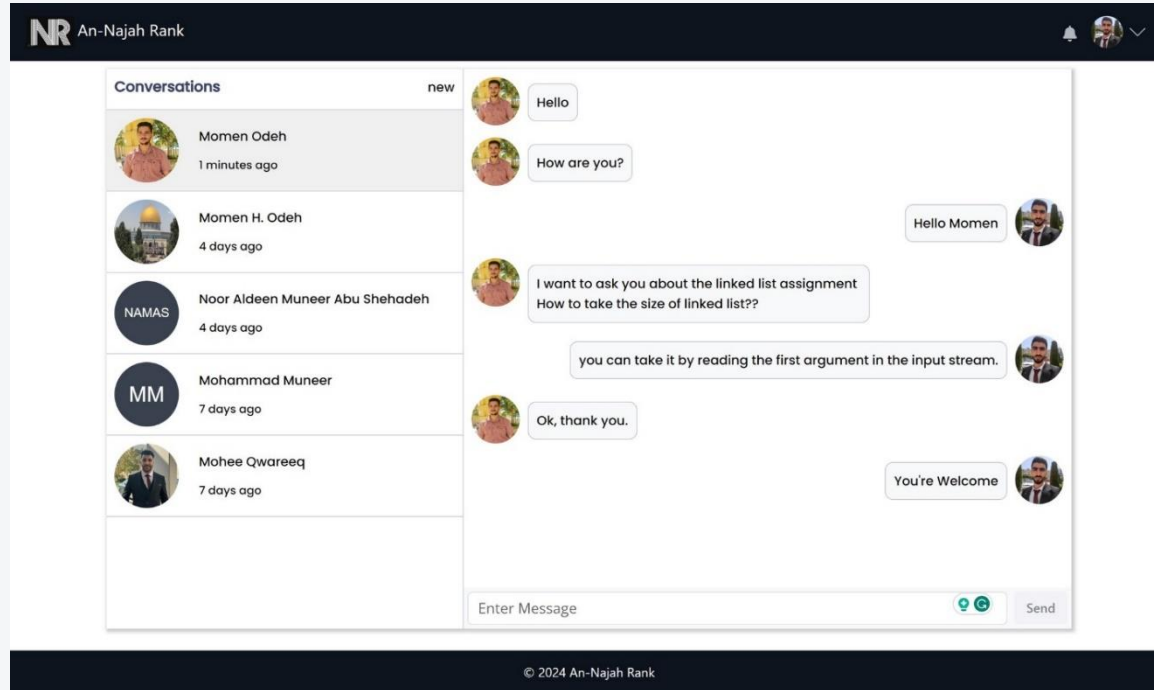
Save Changes

An-Najah Rank

© 2024 An-Najah Rank

User Features


Chatting:



User Features

Notifications:


NR An-Najah Rank




Momen Odeh
11923929
student
s11923929@stu.najah.edu

Edit Profile

Student Statistics


	Easy	3/3	100.00%
	Medium	0/1	0%
	Hard	0/0	0%

Courses



Computer Programming

Momen H. Odeh Noor Aldeen Abu Shehadeh



Data Structure

Momen H. Odeh Noor Aldeen Abu Shehadeh

[Show all Courses](#)

New challenge added to contest in Data Structure course

Notifications

New challenge added to contest in Data Structure course

3 minutes ago

New contest added to Data Structure course

9 days ago

New challenge added to contest in Computer Programming course

9 days ago

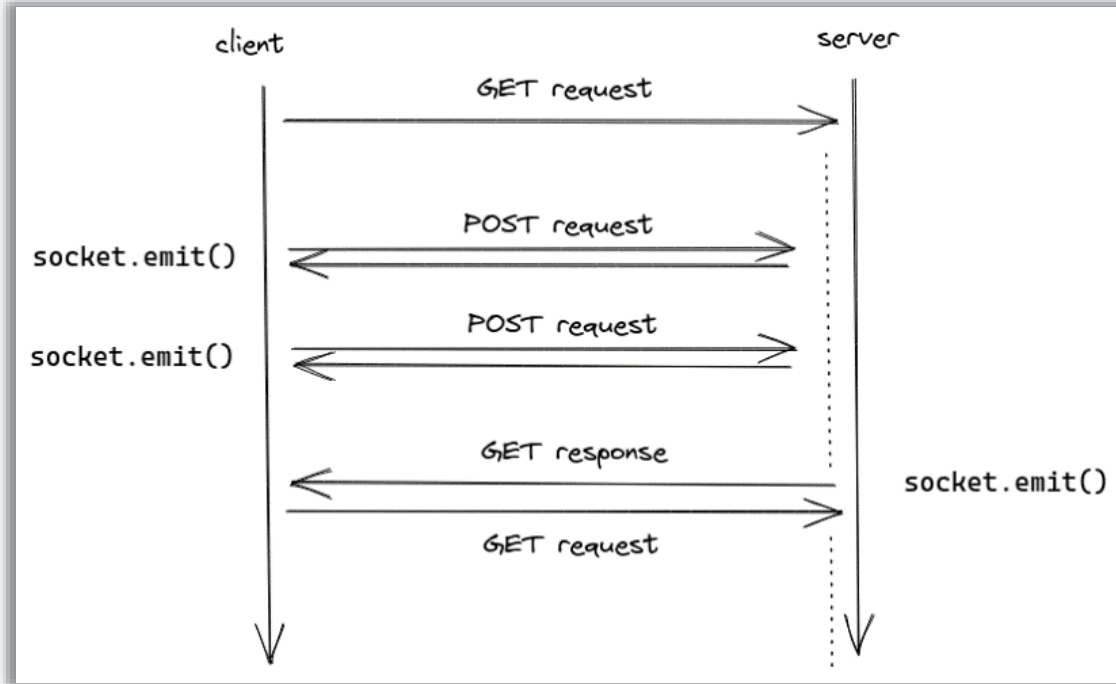
New challenge added to contest in Computer Programming course

9 days ago

New challenge added to contest in Computer Programming course





9 days ago

Socket IO:



Professor Features

NR An-Najah Rank



[administration](#) > [courses](#)

Administration

Manage Courses





Manage Challenges

Create Course

Course Name	Course Owner	Moderators
Data Structure	Noor Aldeen Abu Shehadeh	Momen H. Odeh
Computer Programming	Momen H. Odeh	Noor Aldeen Abu Shehadeh

Mange Courses

NR An-Najah Rank



[administration](#) > [challenges](#)

Administration

Manage Courses

Manage Challenges

Create Challenge

Challenge Name	Challenge tags	Challenge Owner
Print Linked List In Reverse	data structure	Noor Aldeen Abu Shehadeh
Add Two Numbers		Noor Aldeen Abu Shehadeh
factorial number		Noor Aldeen Abu Shehadeh
prime number		Noor Aldeen Abu Shehadeh

Mange Challenges

Professor Features (cont.)

Course Management:

NR

An-Najah Rank

administration > courses > 10636211 > details

Data Structure

Details

Moderators

Course Students

Manage Contests

Course Number

10636211

Course Name

Data Structure

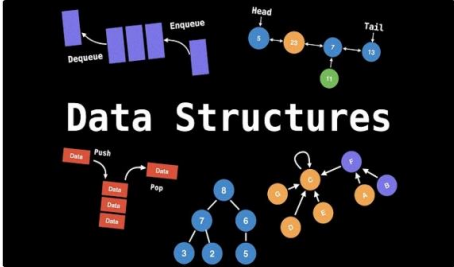
Description

A data structure is a way of organizing and storing data to perform operations efficiently. It defines the relationship between data elements, the operations that can be performed on the data, and the rules for organizing

Background Image

Choose File

No file chosen



Data Structures

Cancel Changes

Save Changes

© 2024 An-Najah Rank

Add students to course

[illegible]

Contest Management :

© 2024 An-Najah Rank

Challenge Management :

Print Linked List In Reverse

Details TestCases

Challenge Difficulty Easy

Specify Language ☒ Java ☒ C ☒ C++ ☒ Python ☒ JavaScript ☐ Regex

Challenge Name Print Linked List In Reverse

Description

Problem Statement

Normal

get data from input screen and build a linked list then print the linked list in reverse

Input Format

Normal

The first line contains an integer , the number of elements in the linked list.
The next lines contain integers for the linked list data separated by space.

Constraints

Normal





Output Format

Normal

Professor Features (cont.)

Manage test cases in challenge

NR An-Najah Rank



[administration](#) > [challenges](#) > [47](#) > [test-cases](#)













Print Linked List In Reverse

Details

TestCases

Add Test Case

*** Should add at least one sample test case to enable use this challenge.**

Order	Input	Output	Is Sample	Strength	
0	3 1 2 3	3 2 1		0	 
1	5 3 7 2 12 10	10 12 2 7 3		10	 
2	6 45 8 9 7 12 0	0 12 7 9 8 45		10	 
3	1 5	5		10	 

© 2024 An-Najah Rank

Professor Features (cont.)

Add new test case when there is a submission for challenge

The screenshot shows the 'Add Test Case' modal in the An-Najah Rank administration system. The modal is titled 'Add Test Case' and contains the following elements:

- Warning:** A red message states: '* This challenge is used in courses and there is student submit code please choose the contest in course who want to run this test case on it.'
- Selection:** A checkbox labeled 'contest 81 - Linked List in course 10636211 - Data Structure.' is checked.
- Strength:** A dropdown menu is set to '10', with a 'Sample' checkbox next to it.
- Input:** A text area containing the following content:

```
1 1
2 2
```
- Output:** A text area containing the following content:

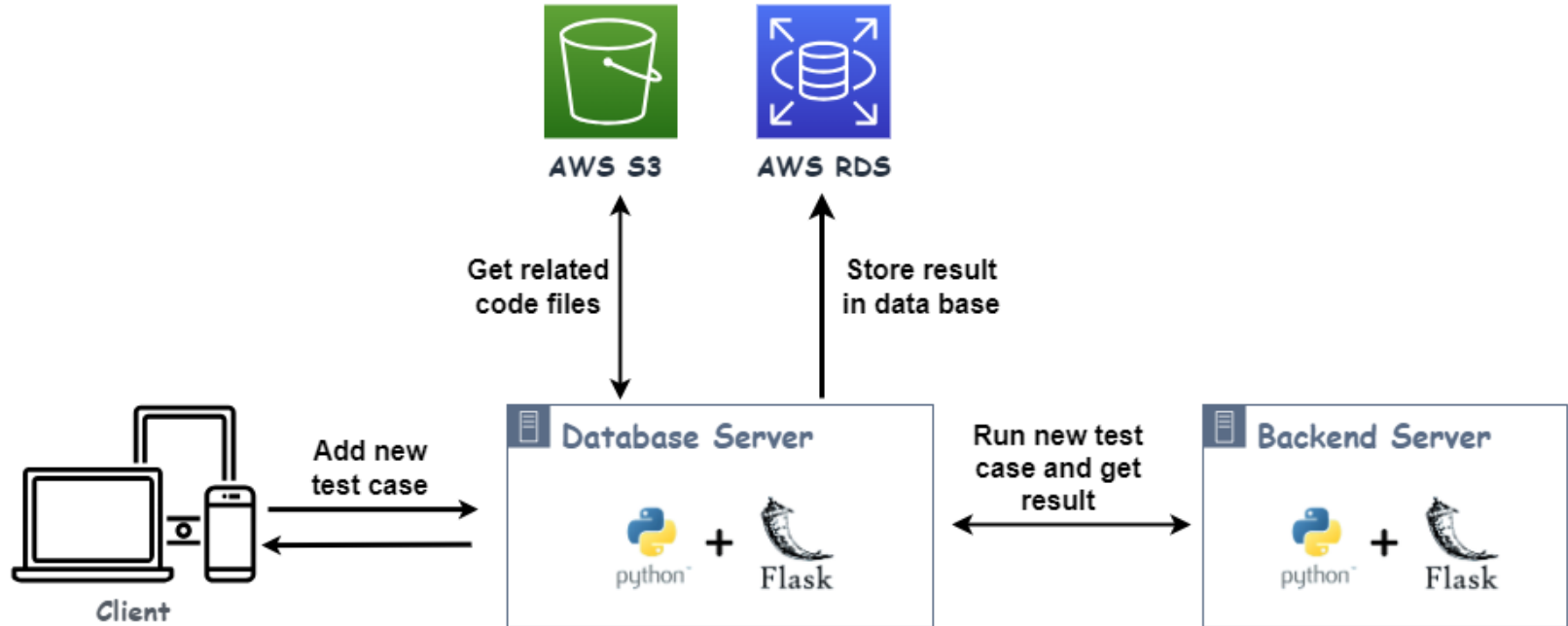
```
1 2
```
- Buttons:** A 'Save' button at the bottom right and an 'Add Test Case' button on the right side of the background interface.

In the background, a table titled 'Print Linked Li' is visible, showing a list of test cases with columns for 'Order' and 'Input'.

Order	Input
0	3 1 2
1	5 3 7
2	6 4 5 8
3	1 5
4	1 2


© 2024 An-Najah Rank






Add new test case when there is a submission for challenge (cont.)





Professor Features (cont.)

Profile:

 An-Najah Rank


    




 **Noor Aldeen Abu Shehadeh**


1945

professor

 anooraldeen9@gmail.com

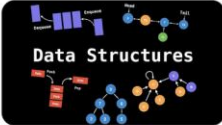
Edit Profile

 Courses



Computer Programming

Momen H. Odeh Noor Aldeen Abu Shehadeh



Data Structure

Momen H. Odeh Noor Aldeen Abu Shehadeh

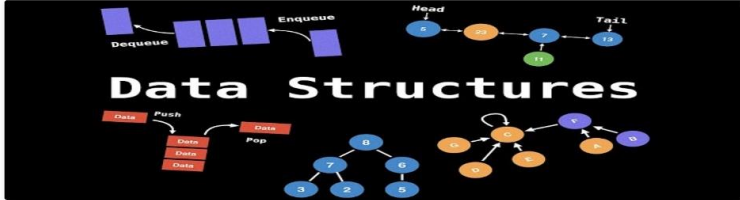
[Show all Courses](#)

© 2024 An-Najah Rank


Professor Features (cont.)


Course View:

NR An-Najah Rank

The banner features the title "Data Structures" in large white font. Above it, there are diagrams for a queue (labeled "Dequeue" and "Enqueue" with blue blocks) and a linked list (labeled "Head" and "Tail" with nodes containing numbers). Below the title, there are diagrams for a stack (labeled "Push" and "Pop" with red blocks) and a tree structure (labeled "A", "B", "C", "D", "E", "F", "G", "H" with nodes containing numbers).

[courses](#) > 10636211

 **Data Structure**

 **Description**

A data structure is a way of organizing and storing data to perform operations efficiently. It defines the relationship between data elements, the operations that can be performed on the data, and the rules for organizing the data. Different types of data structures serve various purposes, and their selection depends on the specific requirements of a task or problem.

Contests

Course Students

Contests [Add Contest](#)

Linked List

10 days 0 hours 56 minutes 22 seconds

Solved Rate: 5.66% max Score: 25

[View Contest](#)

Tree

Start After 9 days 23 hours 22 minutes 22 seconds

Solved Rate: 0%

[View Contest](#)

© 2024 An-Najah Rank

Professor Features (cont.)

Contest View:

NR

An-Najah Rank

courses > 10636211 > contests > 81

Linked List

Description

A **linked list** is a linear data structure where elements, called nodes, are connected through pointers, forming a sequence. Each node contains data and a reference to the next node in the sequence.

Remaining time

10
days

0
hours

54
minutes

52
seconds

</> Challenges

Print Linked List In Reverse

Difficulty: **Easy** Success Rate: **5.66 %** Max Score: **25**

View Challenge

© 2024 An-Najah Rank

Professor Features (cont.)

Challenge submission View:

NR

An-Najah Rank

courses > 10636211 > contests > 81 > challenges > 47 > submissions

Print Linked List In Reverse

Problem

Submissions

Leaderboard

Type student name

Calculate Similarity

Name	Date	Score ▼ ▲	Similarity ▼ ▲		
Noor Aldeen Muneer Abu Shehadeh	Wed, 10 Jan 2024 16:15:26 GMT	<div>25</div> <div>25</div>	71%	<div>View Submissions</div>	<div>View Similarity</div>
Mohammad Muneer	Wed, 10 Jan 2024 17:40:29 GMT	<div>25</div> <div>25</div>	64%	<div>View Submissions</div>	<div>View Similarity</div>
Momen Odeh	Wed, 10 Jan 2024 17:40:48 GMT	<div>25</div> <div>25</div>	30%	<div>View Submissions</div>	<div>View Similarity</div>

Similarity data ready for

submissions in Data Structure course

© 2024 An-Najah Rank

Professor Features (cont.)

Challenge student submissions View:

NR

An-Najah Rank

courses > 10636111 > contests > 80 > challenges > 49 > submissions > manual-mark > 11923929

Submission 2

Submission 1

Submission Details

Submitted at: 1/10/2024, 9:23:44 PM

Score out of 100: 100

Save Changes

Submitted Code

Language: java

```
1 import java.io.*;
2 import java.util.*;
3
4 class Main {
5
6     public static void main(String[] args) {
7         Scanner in = new Scanner(System.in);
8         int num = in.nextInt();
9         int res = 1;
10        for(int i=1; i<=num; i++)
11        {
12            res*= i;
13        }
14        System.out.println(res);
15    }
16 }
```

TestCase 0 (0.0%) ✓

TestCase 1 (33.3%) ✓

TestCase 2 (33.3%) ✓

TestCase 3 (33.3%) ✓

Congratulations, you passed the sample test case.

Input (stdin)

1

Your Output (stdout)

1

Expected Output

1

© 2024 An-Najah Rank

Professor Features (cont.)

Challenge student similarity View:



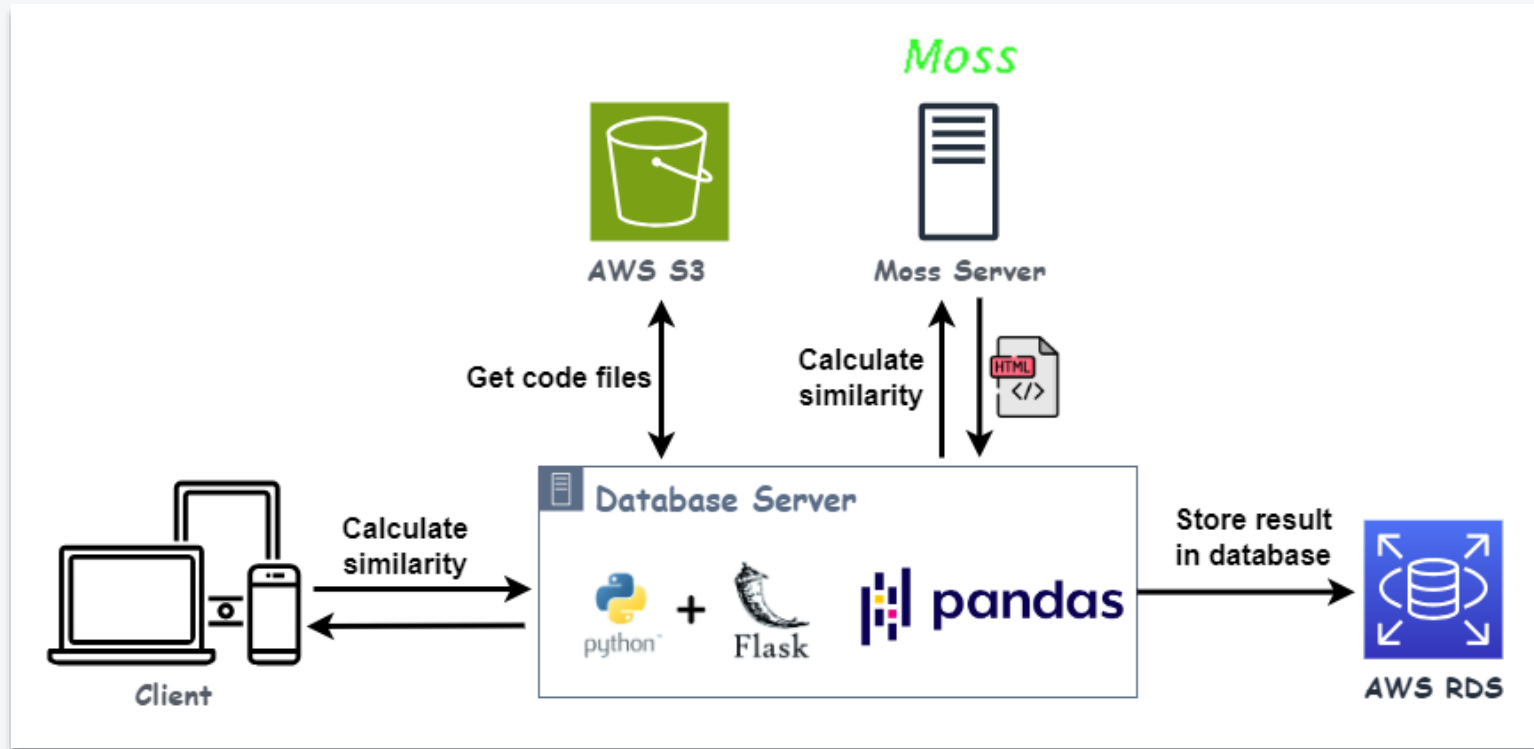
[courses](#) > [10636211](#) > [contests](#) > [81](#) > [challenges](#) > [47](#) > [submissions](#) > [code-similarity](#) > [11923513](#)

Code Similarity Summary

Noor_Aldeen_Muneer_Abu_Shehadeh-11923513 (71%)	Mohammad_Muneer-11235499 (58%)	...
<pre>10 struct Node* insertNode(struct Node* head, int data) { 11 struct Node* newNode = (struct Node*)malloc(sizeof(struct 12 newNode->data = data; 13 newNode->next = NULL; 14 15 if (head == NULL) { 16 return newNode; 17 } 18 19 struct Node* current = head; 20 while (current->next != NULL) { 21 current = current->next; 22 } 23 24 current->next = newNode; 25 return head; 26 } 27 void printReverse(struct Node* head) {</pre>	<pre>9 struct Node* insertNode(struct Node* head, int data) { 10 struct Node* newNode = (struct Node*)malloc(sizeof(struct 11 newNode->data = data; 12 newNode->next = head; 13 return newNode; 14 } 15 16 void printReverse(struct Node* head) { 17 if (head == NULL) { 18 return; 19 } 20 printf("%d ", head->data); 21 printReverse(head->next); 22 } 23 24 int main() { 25 int size; 26 scanf("%d", &size);</pre>	


Mohammad_Muneer-11235499 (58%)	...
Mohammad_Muneer-11235499 (58%)	
Momen_Odeh-11923929 (43%)	


Challenge student similarity View (cont.):





Student Features


Student Profile:



 **Momen Odeh**

 **# 11923929**

 **student**

 **s11923929@stu.najah.edu**

Edit Profile

60.00%

Solved

Easy

3/4

75.00%

Medium

0/1


0%

Hard

0/0


0%

Student Statistics



Computer Programming

Momen H. Odeh Noor Aldeen Abu Shehadeh



Data Structure

Momen H. Odeh Noor Aldeen Abu Shehadeh

Show all Courses

</> Latest Challenges

factorial number

Difficulty: **Easy** Success Rate: **0%** Max Score: **25**

Solve Challenge

Student Features

Solving challenge:

NR An-Najah Rank

[courses > 10636211](#) > [contests > 81](#) > [challenges > 47](#) > [problem](#)

Print Linked List In Reverse

Problem Submissions Leaderboard

-

Input Format

The first line contains an integer , the number of elements in the linked list.

The next lines contain an integers for the linked list data separated by space.

Constraints

-

Output Format

an integers of reverse linked list data separated by space.

Simple Input 0

```
3
1 2 3
```

Sample Output 0

```
3 2 1
```

Explanation 0

this is a sample of reverse print linked list.

Dark mode: ☐ C

Dark mode: ☐ C

```
1 #include <stdio.h>
2 #include <string.h>
3 #include <math.h>
4 #include <stdlib.h>
5
6 struct Node {
7     int data;
8     struct Node* next;
9 };
10 struct Node* insertNode(struct Node* head, int data) {
11     struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
12     newNode->data = data;
13     newNode->next = NULL;
14
15     if (head == NULL) {
16         return newNode;
17     }
18
19     struct Node* current = head;
20     while (current->next != NULL) {
21         current = current->next;
22     }
```

Run Code Submit Code

TestCase 0 ✓

Congratulations, you passed the sample test case.

Input (stdin)

```
3
1 2 3
```

Your Output (stdout)

```
3 2 1
```

Expected Output

```
3 2 1
```

Code operation:

Language	Compiler/Interrupter
C/C++	
Java	
Python	
JavaScript	

Student Features

Student submission:

NR

An-Najah Rank

courses > 10636211 > contests > 81 > challenges > 47 > submissions

Print Linked List In Reverse

Problem

Submissions

Leaderboard





Problem	Language	Time	Result	Score	
Print Linked List In Reverse	c	Wed, 10 Jan 2024 22:45:20 GMT	Wrong Answer ❌	0	<div>View Result</div>
Print Linked List In Reverse	c	Wed, 10 Jan 2024 22:46:53 GMT	Accepted ✅	25	<div>View Result</div>

© 2024 An-Najah Rank

Student Features

Student submission (cont.):

NR An-Najah Rank



[courses > 10636211](#) > [contests > 81](#) > [challenges > 47](#) > [submissions > 26](#)

Print Linked List In Reverse

Problem

Submissions

Leaderboard

Submission Details

Submitted at: 1/10/2024, 6:15:26 PM
Score: 25

✓ Test Case #0

✓ Test Case #1

✓ Test Case #2

✓ Test Case #3

Submitted Code

Language: c

Open in editor

```
1 #include <stdio.h>
2 #include <string.h>
3 #include <math.h>
4 #include <stdlib.h>
5
6 struct Node {
7     int data;
8     struct Node* next;
9 };
10 struct Node* insertNode(struct Node* head, int data) {
11     struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
12     newNode->data = data;
13     newNode->next = NULL;
14
15     if (head == NULL) {
16         return newNode;
17     }
18
19     struct Node* current = head;
20     while (current->next != NULL) {
21         current = current->next;
```

© 2024 An-Najah Rank

Admin Features

NR

An-Najah Rank

admin > professors-requests

Welcome Back

Professors RequestsProfessorsStudentsSubmissions

professors request

Type Professor Name

Professor Name	University Number	Email		
Noor Aldeen Abu Shehadeh	1945	anooraldeen9@gmail.com	✓	✗
Momen H. Odeh	11072	momen.odeh74@gmail.com	✓	✗

pending professors

NR

An-Najah Rank

admin > submissions

Welcome Back

Professors RequestsProfessorsStudentsStatistics

Submissions

Type Student Name

Student Name	University Number	Total Submission	Total Success Submission	Rate
Noor Aldeen Muneer Abu Shehadeh	11923513	4	4	100.00%
Momen Odeh	11923929	4	3	75.00%
Mohammad Muneer	11235499	1	1	100.00%

Students statistics

Sample of responsive design


NR

An-Najah Rank

1


Conversations

new




Momen Odeh

1 days ago




Momen H. Odeh

5 days ago




Noor Aldeen Muneer Abu Shehadeh

5 days ago



Mohammad Muneer

8 days ago



Mohee Qwareeq

8 days ago

Enter Message

Send


© 2024 An-Najah Rank

NR


An-Najah Rank

1


>




Hello




How are you?




I want to ask you about the linked list assignment
How to take the size of linked list??




Ok, thank you.



Hello Momen



you can take it by reading the first argument in the input stream.



You're Welcome

Enter Message

Send

© 2024 An-Najah Rank

NR

An-Najah Rank

1

administration > courses > create-course

Create Course

Course Number

Course Name

Description

Background Image

Choose File

No file chosen

Students Excel File

Choose File

No file chosen

* should enter Students Excel File with .xlsx extension

Cancel Changes


Save Changes

© 2024 An-Najah Rank

NR

An-Najah Rank

1



Noor Aldeen Abu Shehadeh


1945

professor

anooraldeen9@gmail.com

Edit Profile


Courses



Computer Programming

Momen H. Odeh

Noor Aldeen Abu Shehadeh



Data Structure

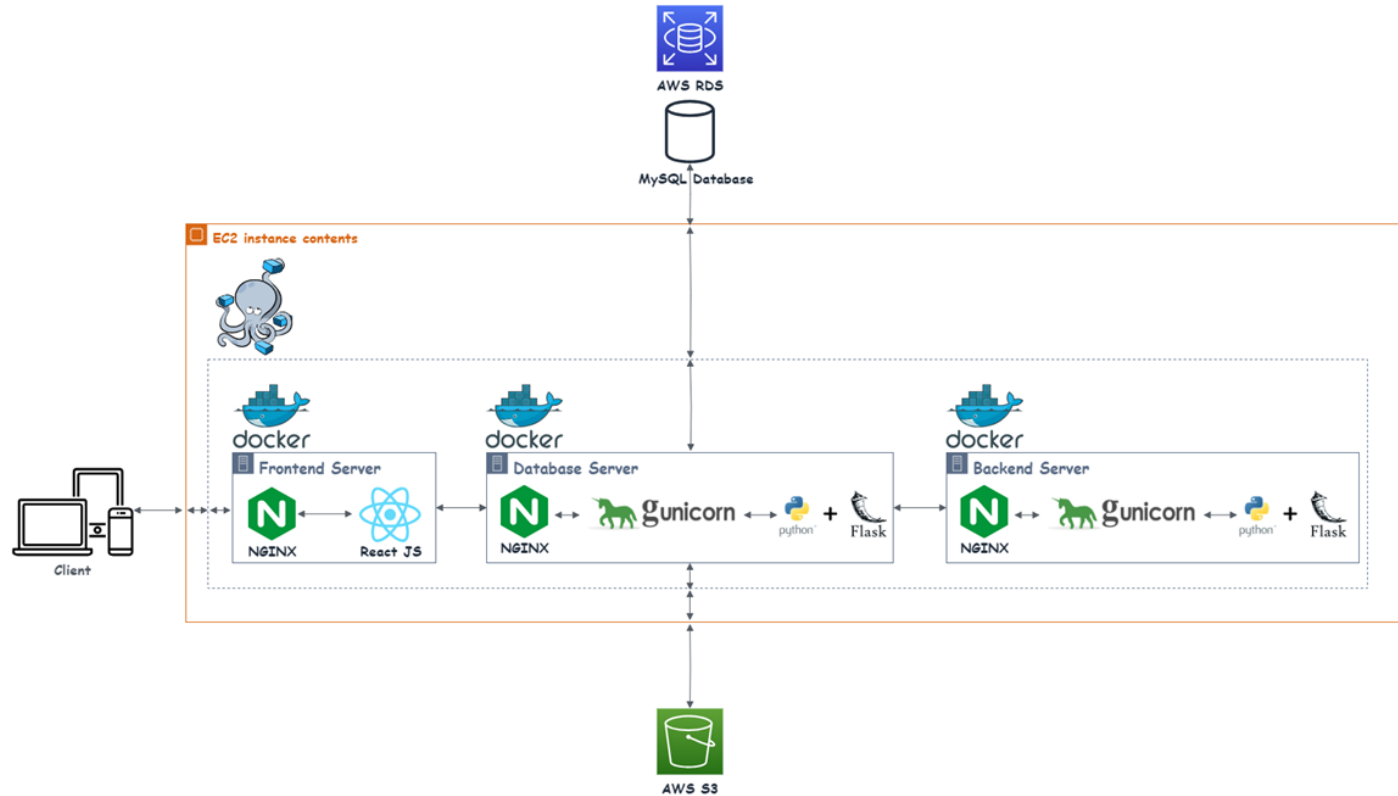
Momen H. Odeh

Noor Aldeen Abu Shehadeh

Show all Courses

© 2024 An-Najah Rank

Deployment Phase



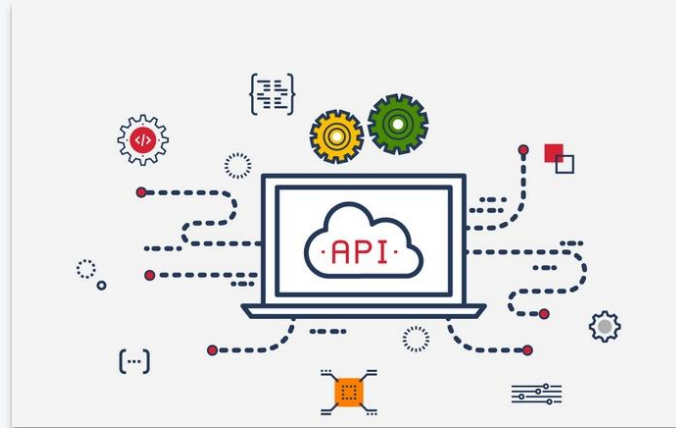
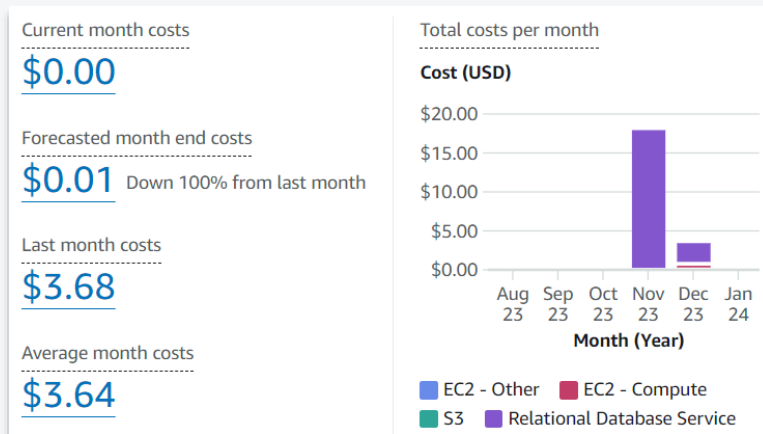
Testing Phase

After implementing the project, we conduct manual testing for all features in the system to ensure that all features work correctly.



Constraints

- ❑ In our AWS environment, not all services come without costs certain services like RDS, EC2, and Elastic IP Addresses require payment.
- ❑ Another challenge we face involves a third-party API we use for similarity calculations. This API is not entirely within our control, and its occasional unavailability may disrupt our similarity calculation processes, potentially affecting the availability of the similarity feature system.



Future Works

- ❑ Support time complexity calculation for the submission code of the challenge.
- ❑ Support creating a challenge related to image processing.



Thank You !