An Najah Rank

Software Graduation Project

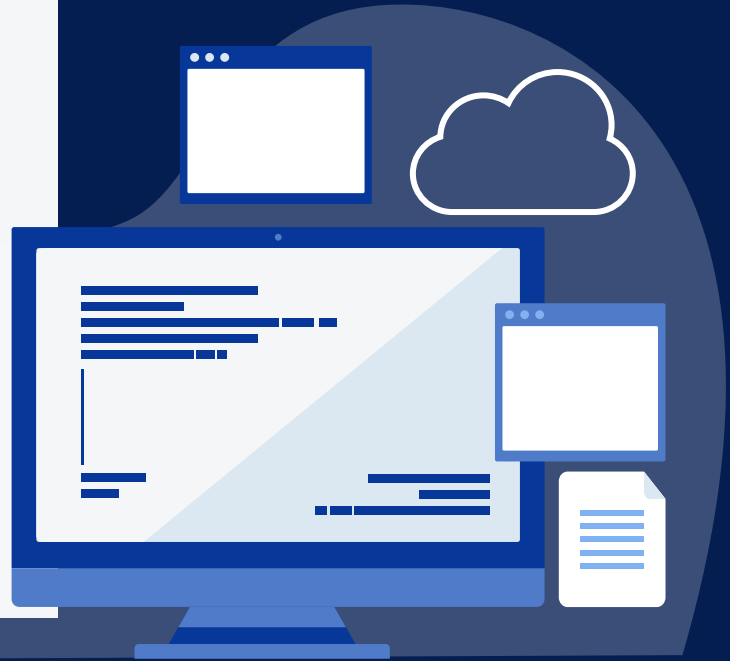
Our Team

Supervisor
Dr. Samer Arandi

Momen Odeh

Noor Aldeen
Abu Shehadeh

Table of contents



What ‘s the problem?

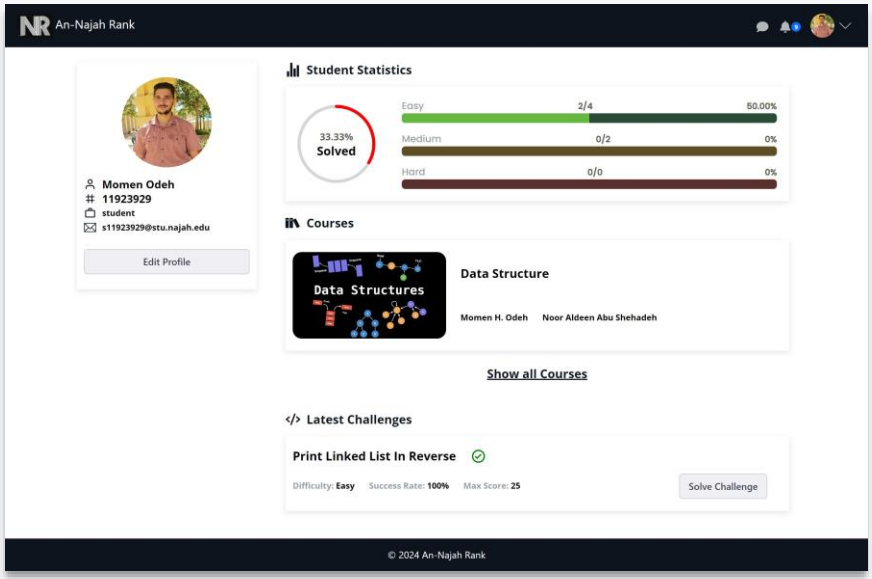
The problem-solving skills are one of the most important skills in the workplace.

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.

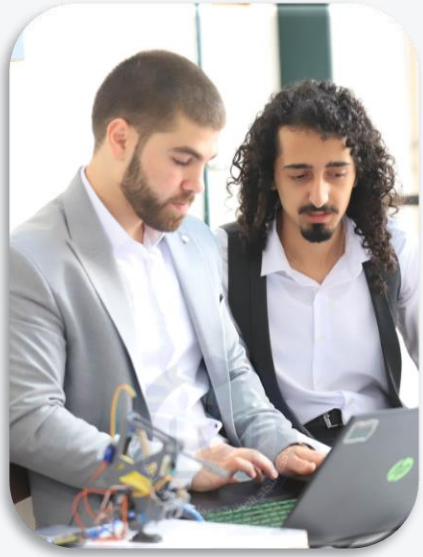
Our Solution

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.

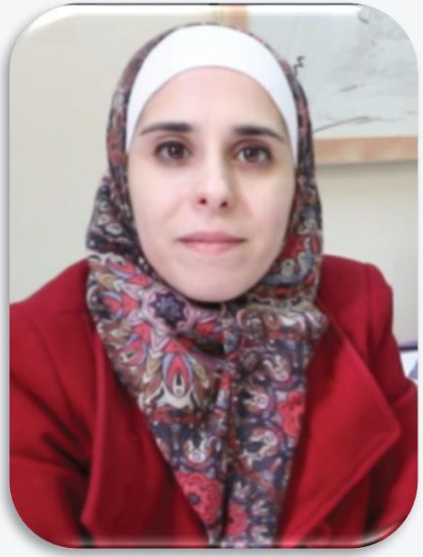


Who use web application ?

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

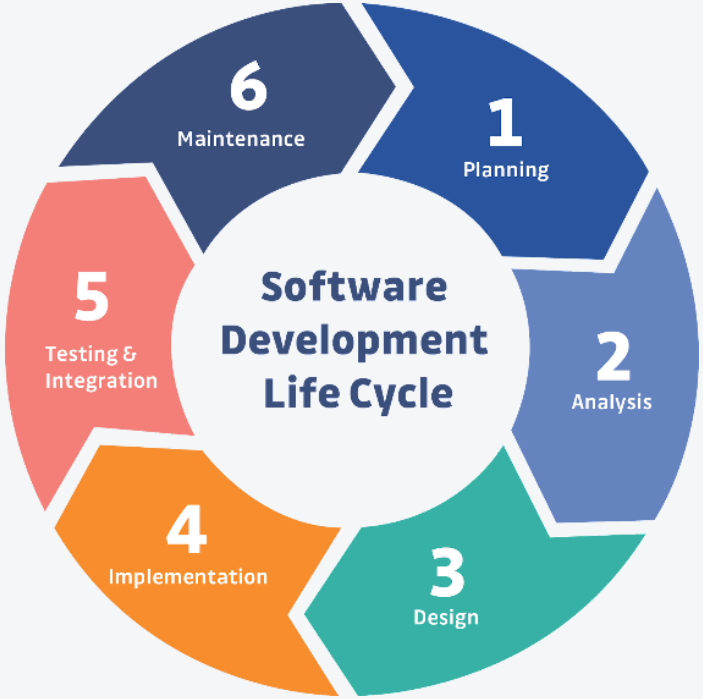
Manual Mark

09

Chatting & Notification

Methodology

Software Development Life Cycle:



Methodology (cont.)

Agile Methodology:

Planning Phase

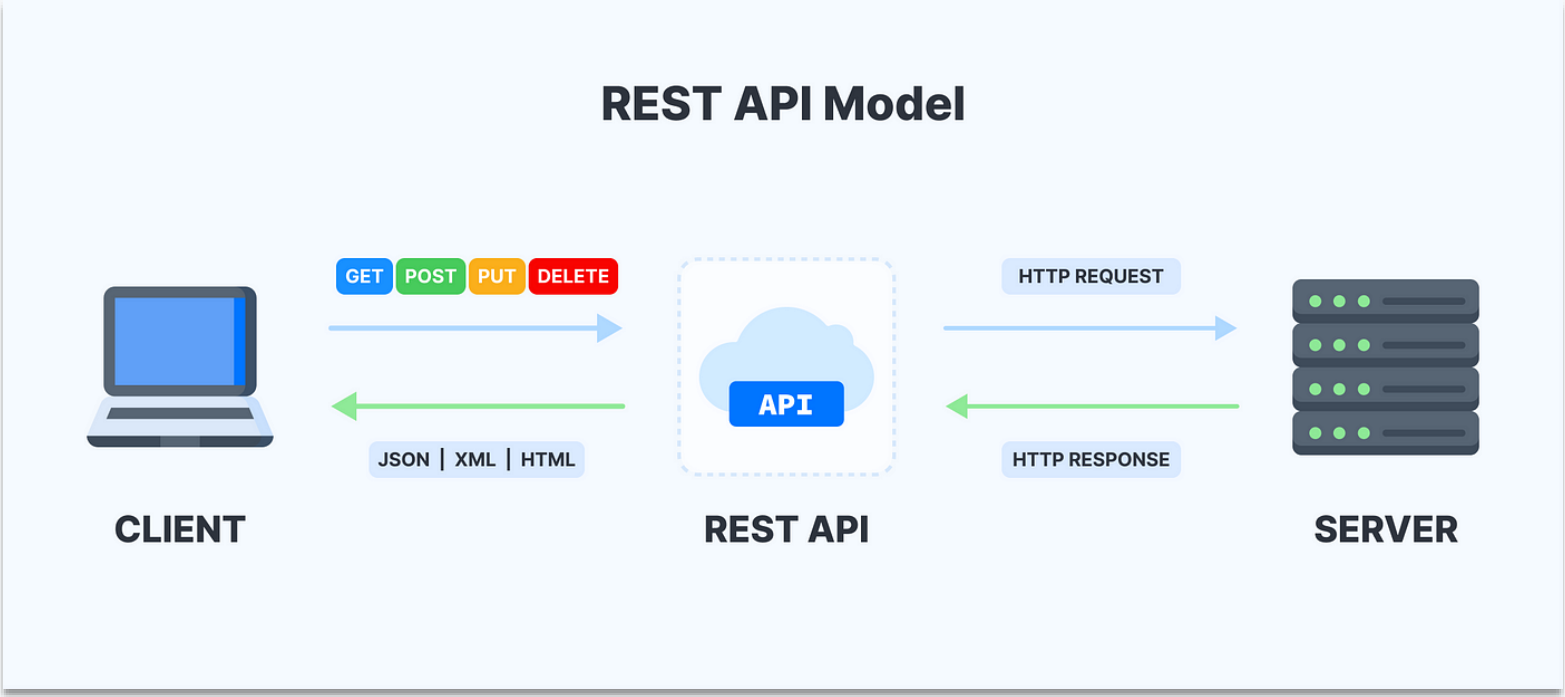
- We met with our supervisor Dr. Samer Arandi.
- We explored various problem-solving websites.
- We are discussing new functionalities to add to the project.

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.

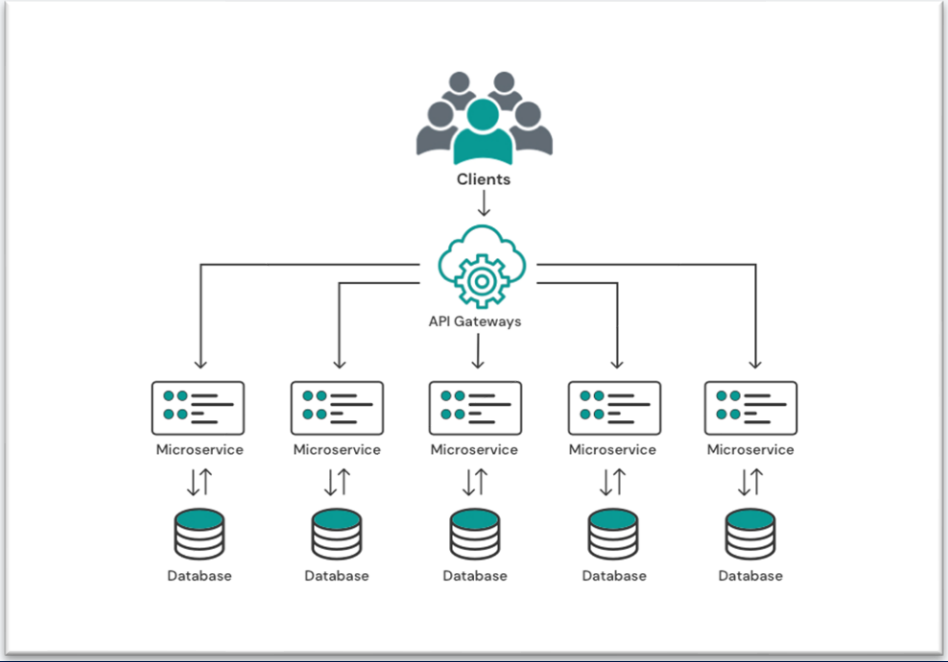
Design Phase

Architectural Style:



Design Phase (cont.)

Architectural Pattern:
Microservice Architectural pattern



Design Phase (cont.)

Frontend libraries:

Backend technologies :

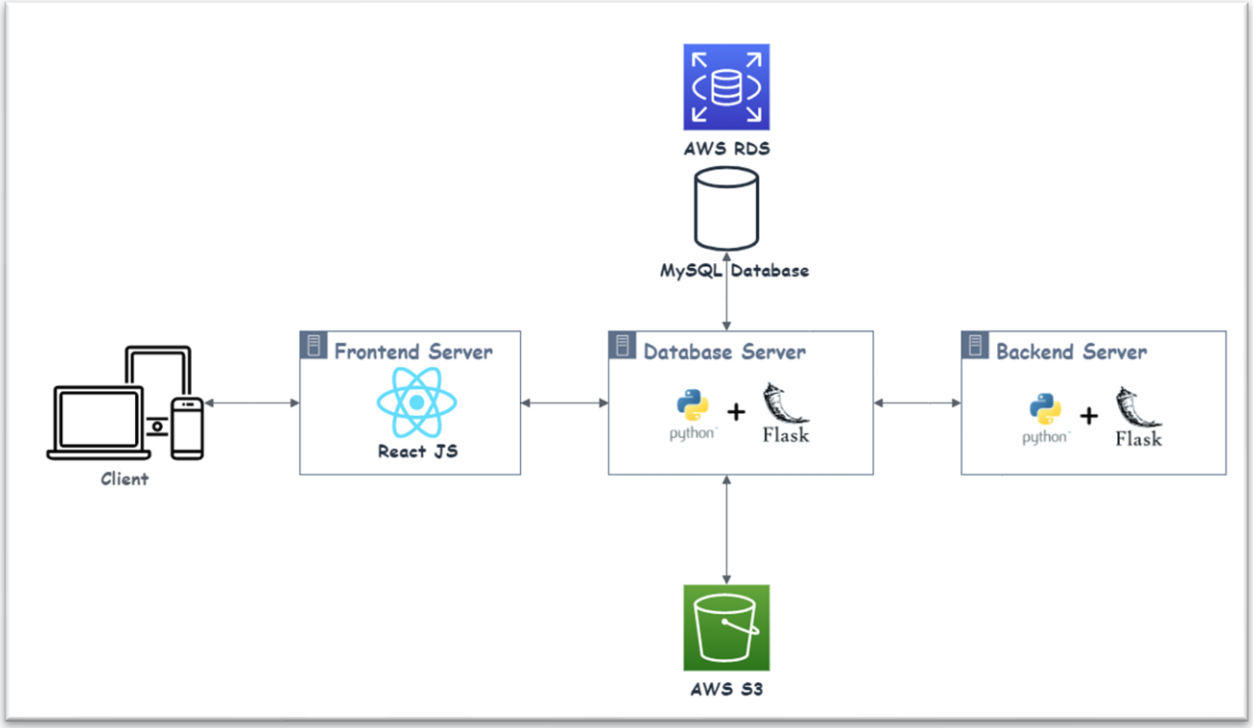
Design Phase (cont.)

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.

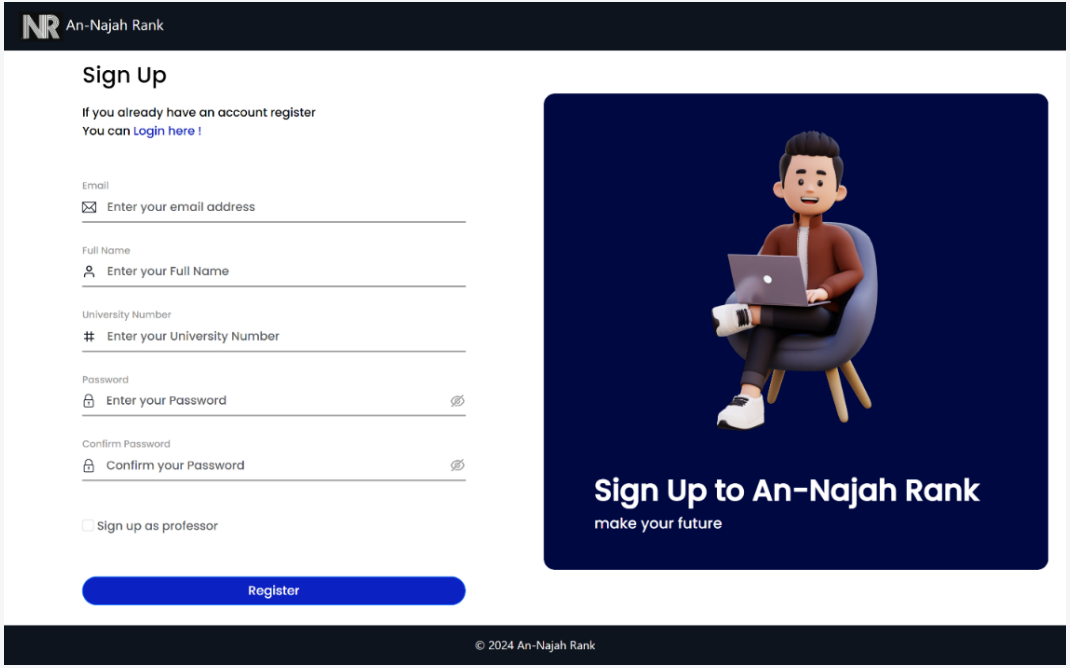
c) **CORS policies**: a mechanism that allows restricted resources on a web page to be accessed from another domain outside the domain from which the first resource

Library used:

JWT CORS flask-mail

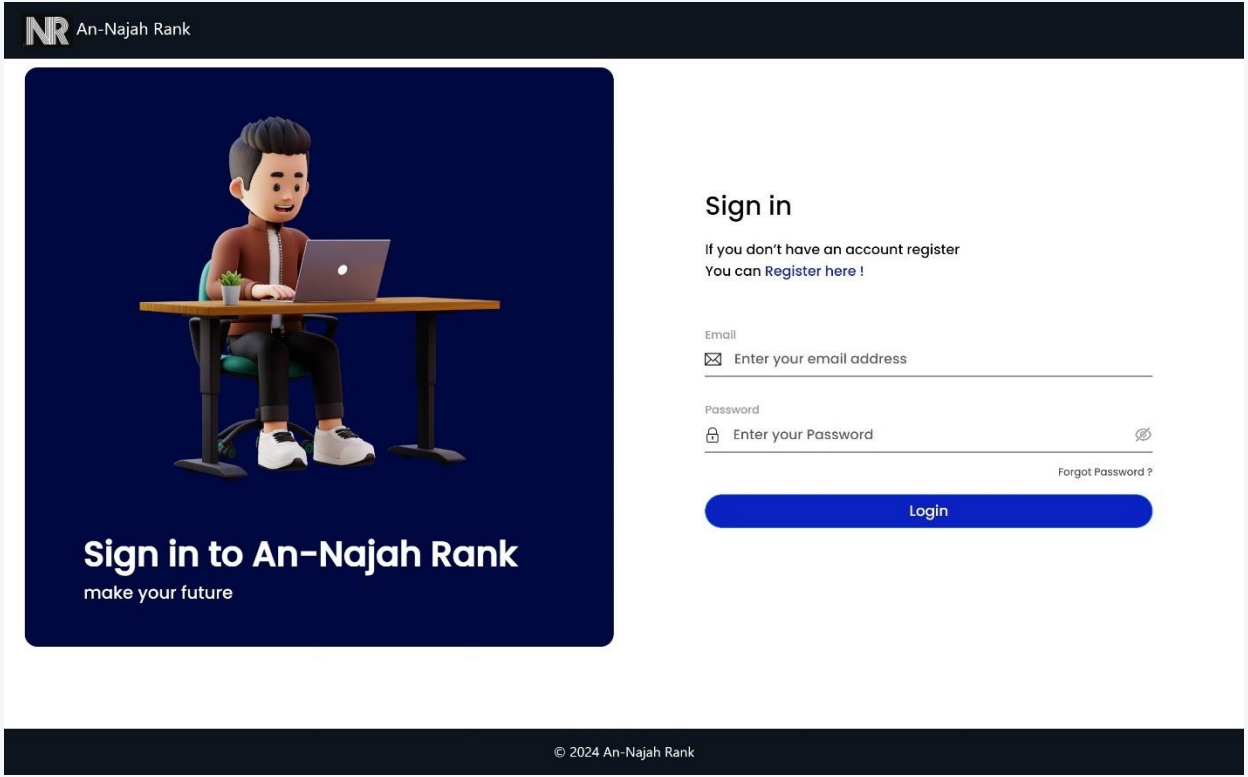
User Features

Registration:



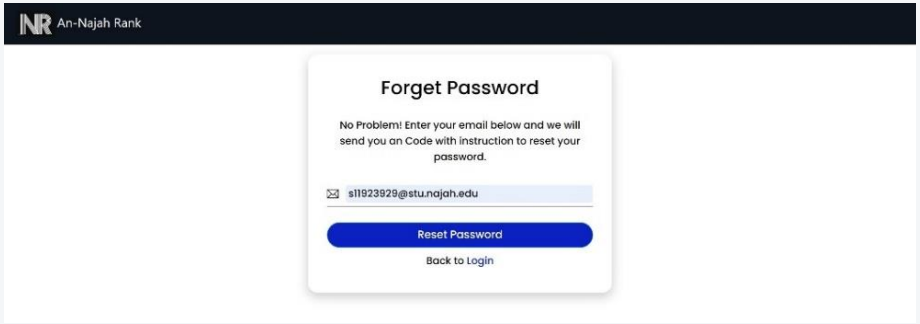
User Features

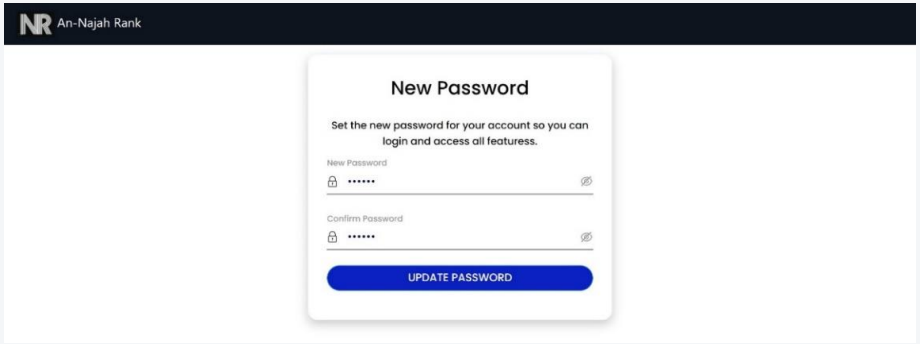
Sign in:



User Features

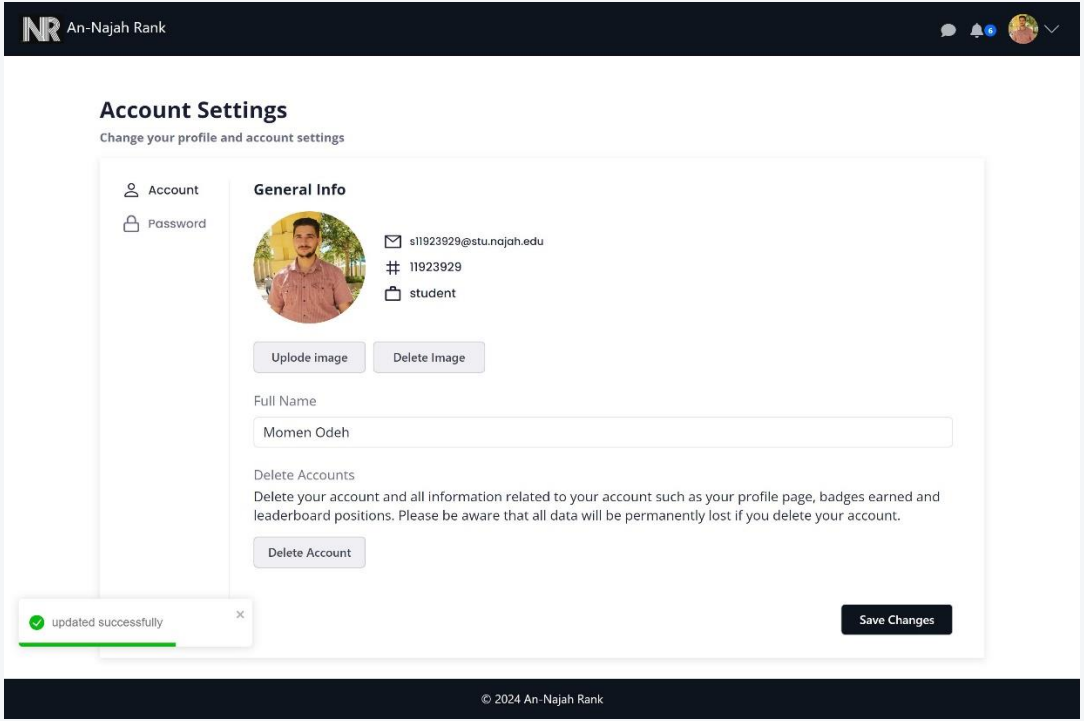
Forget password:





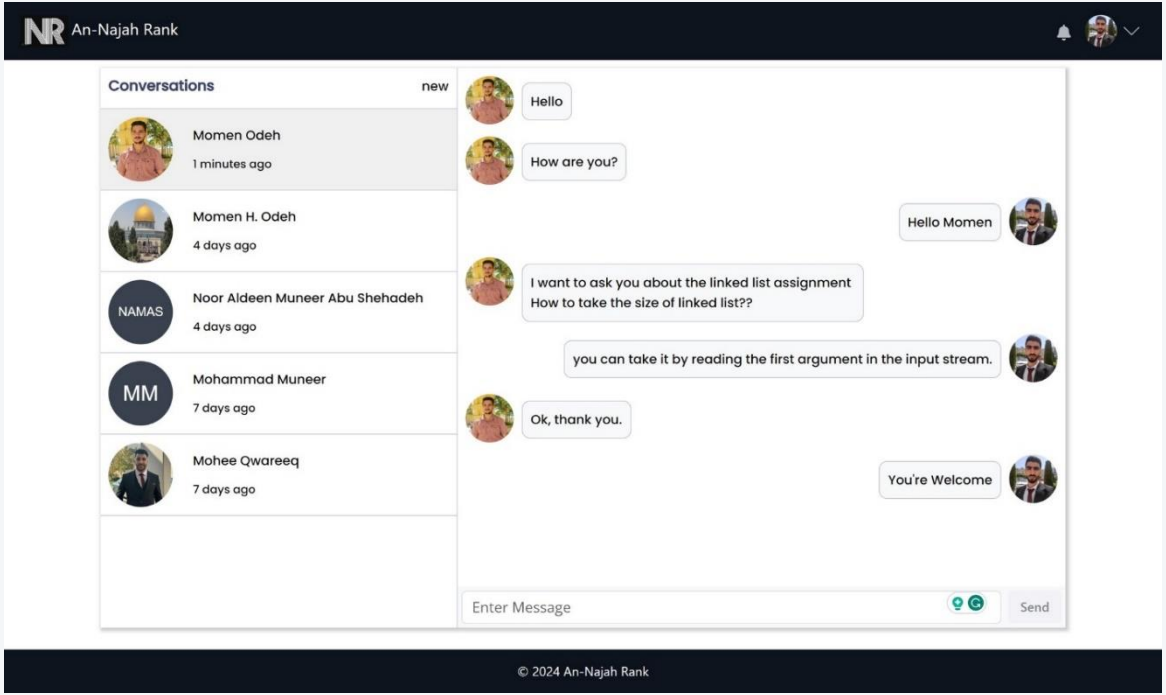
User Features

Account Settings:



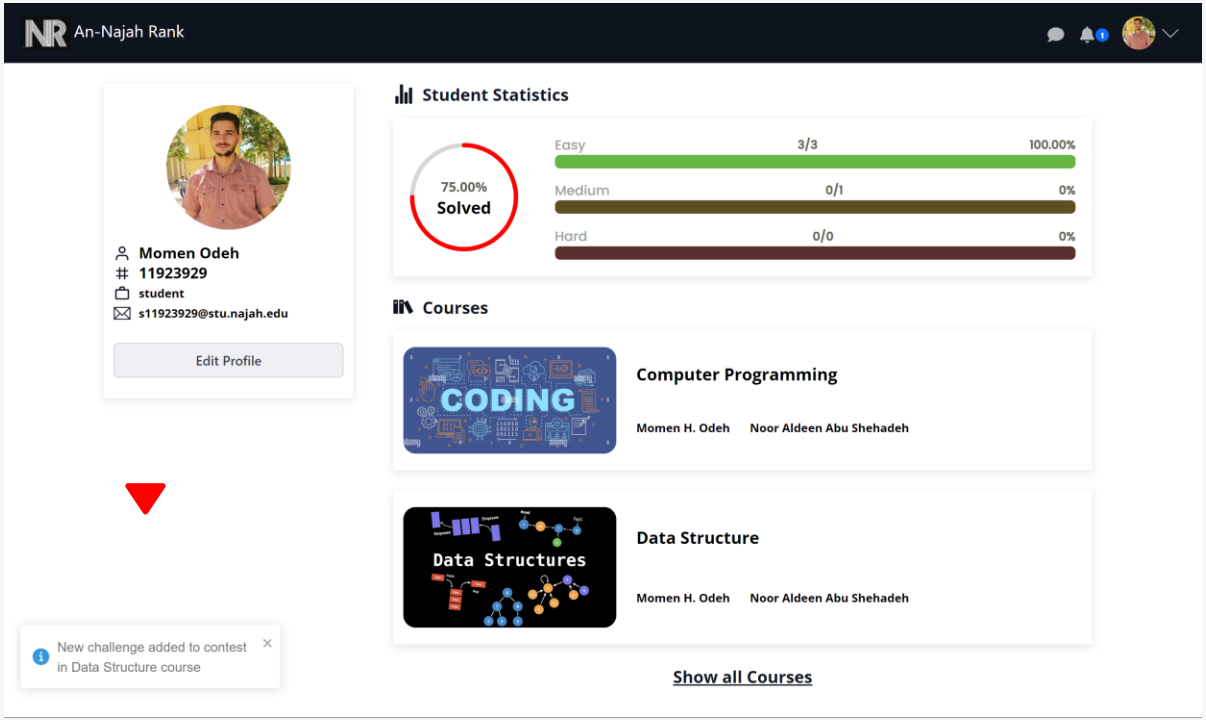
User Features

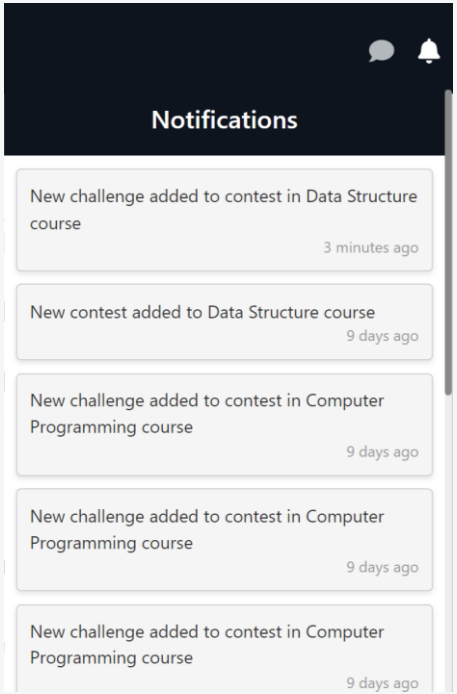
Chatting:



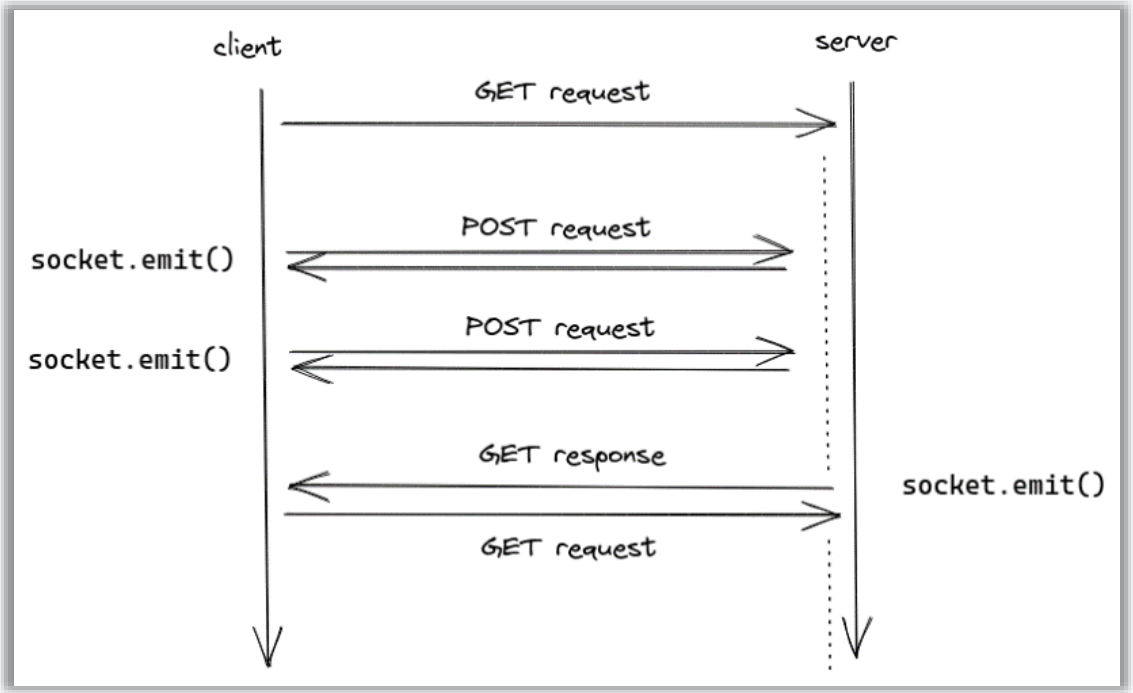
User Features

Notifications:

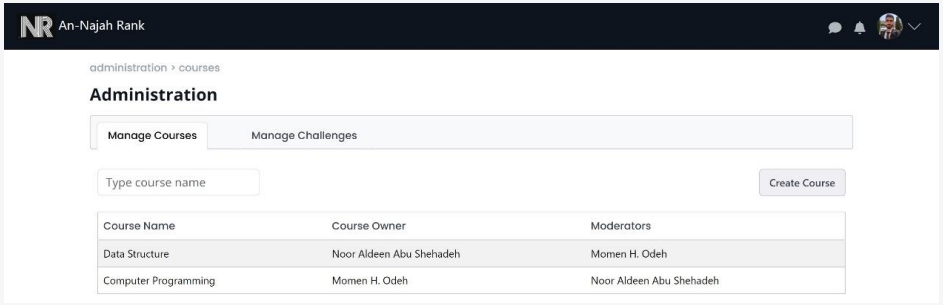




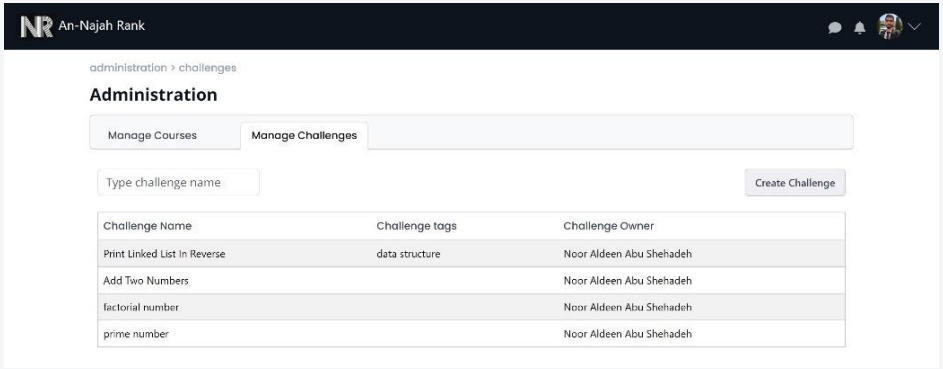
Socket IO:


client
server
GET request
POST request
socket.emit()
POST request
socket.emit()
GET response
socket.emit()
GET request

Professor Features



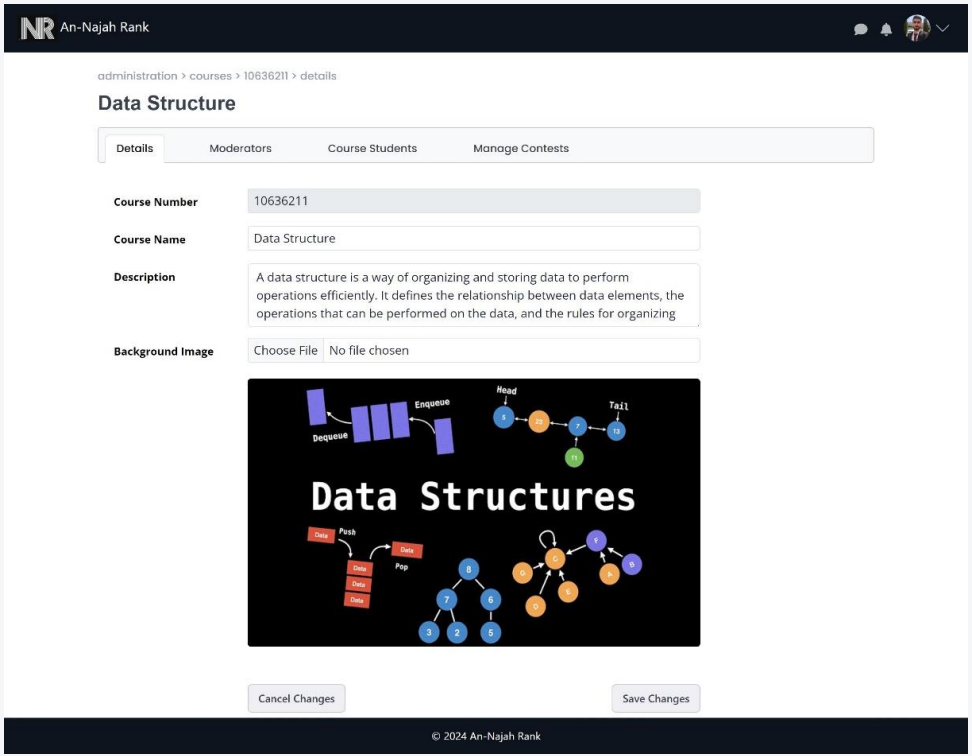
Mange Courses



Mange Challenges

Professor Features (cont.)

Course Management:



Professor Features (cont.)

Add students to course

طباعة قائمة الحضور والغياب

أول 2023 -2024

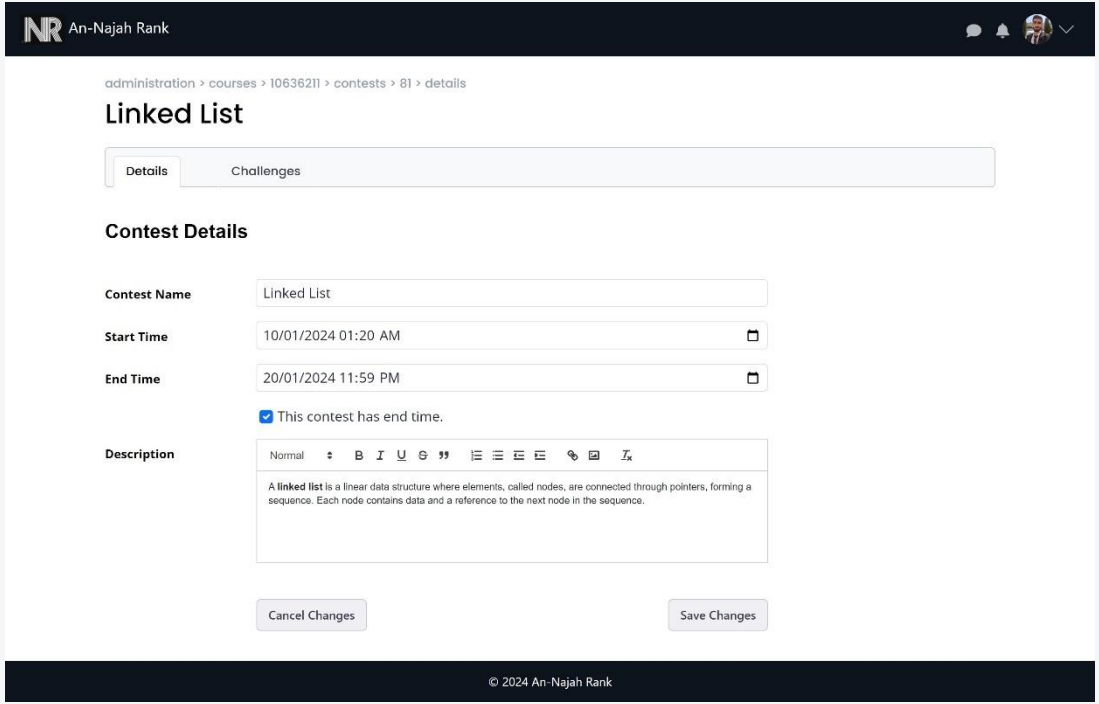
اسم المساق : أنظمة التشغيل الموزعة(10636456/1) اسم المدرس : سامر ناجح ناجي العرندى

الموعد الزمني : احد/ثلاث/خميس 1-2 رقم القاعة : 111170

12	11	10	9	8	7	6	5	4	3	2	1	اسم الطالب	رقم الطالب	رمز الكلية	
															1
															2
															3
															4
															5
															6
															7
															8
															9
															10
															11
															12
															13
															14

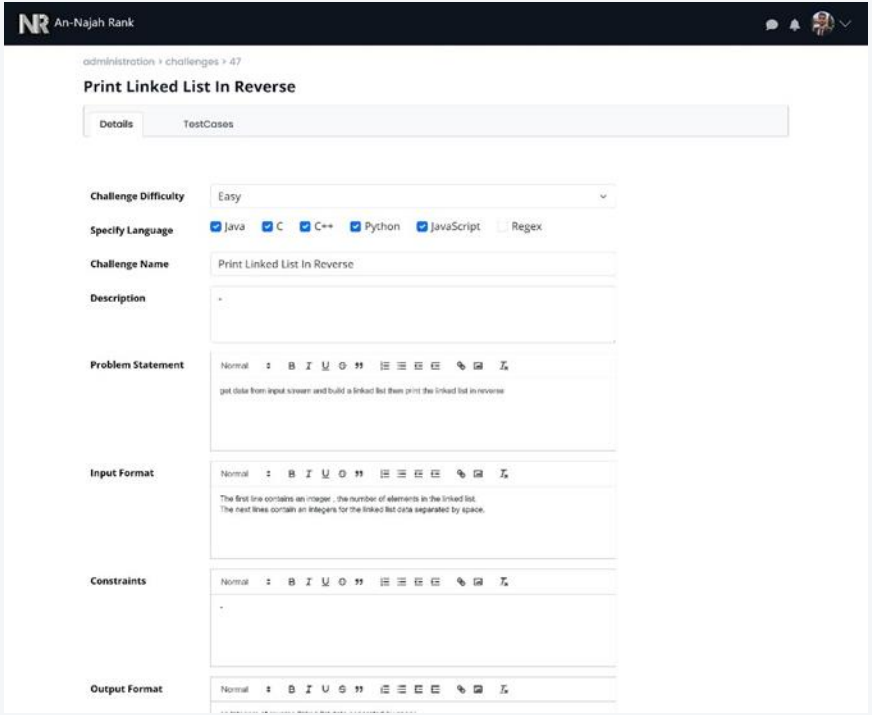
Professor Features (cont.)

Contest Management :



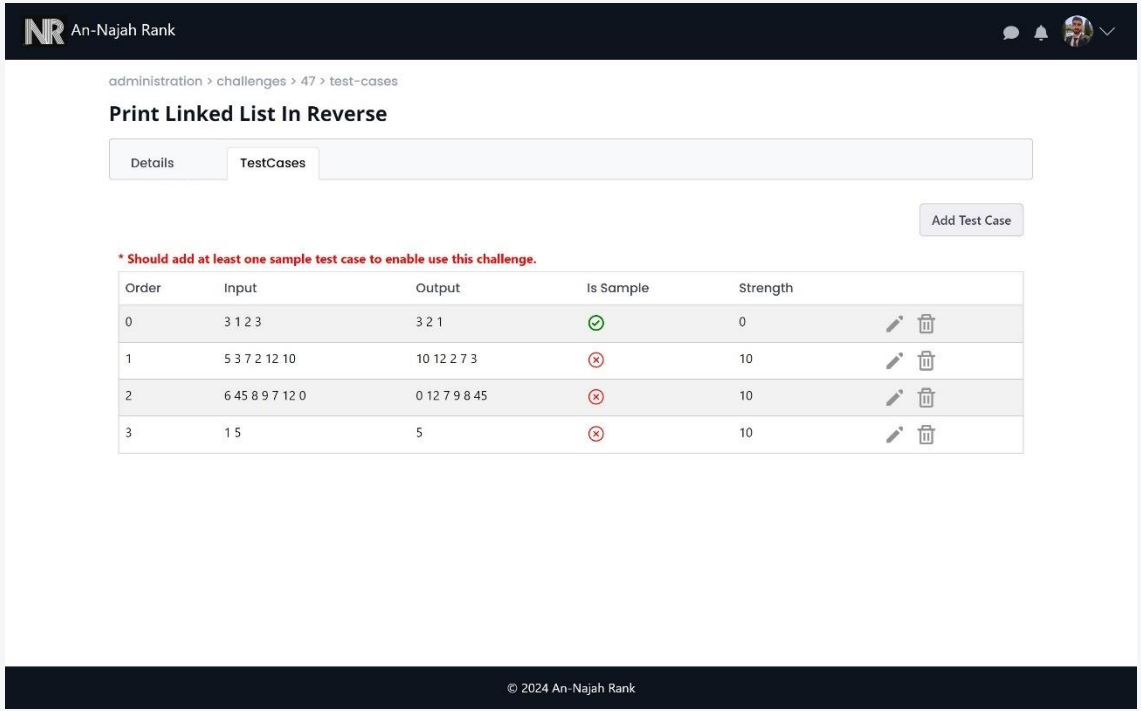
Professor Features (cont.)

Challenge Management :



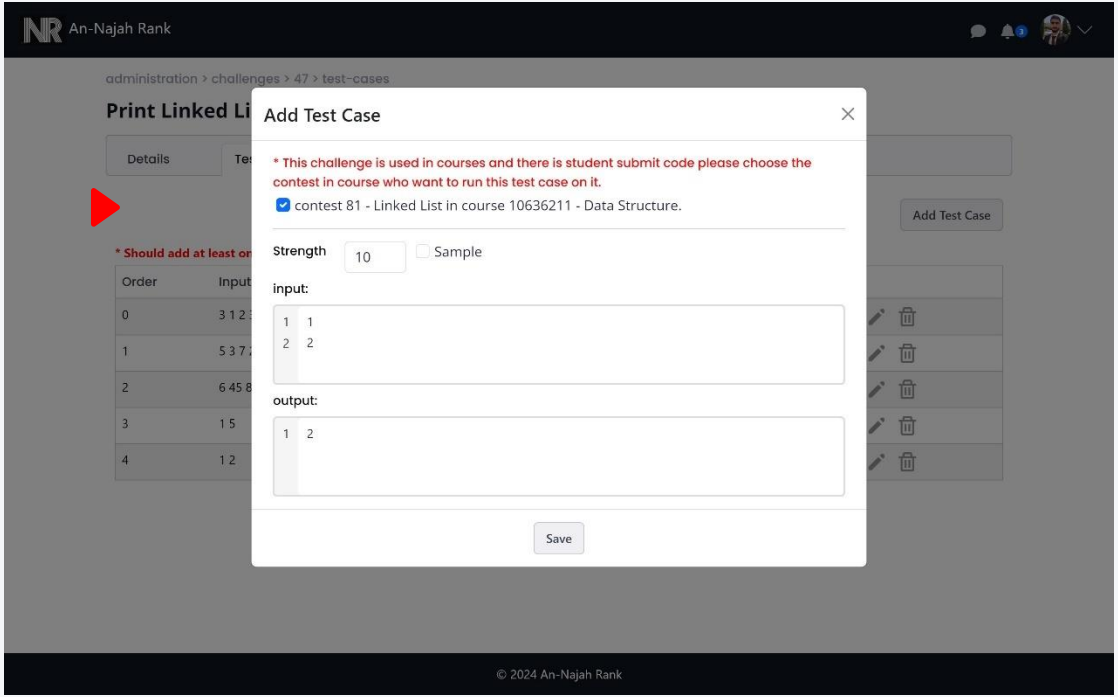
Professor Features (cont.)

Manage test cases in challenge

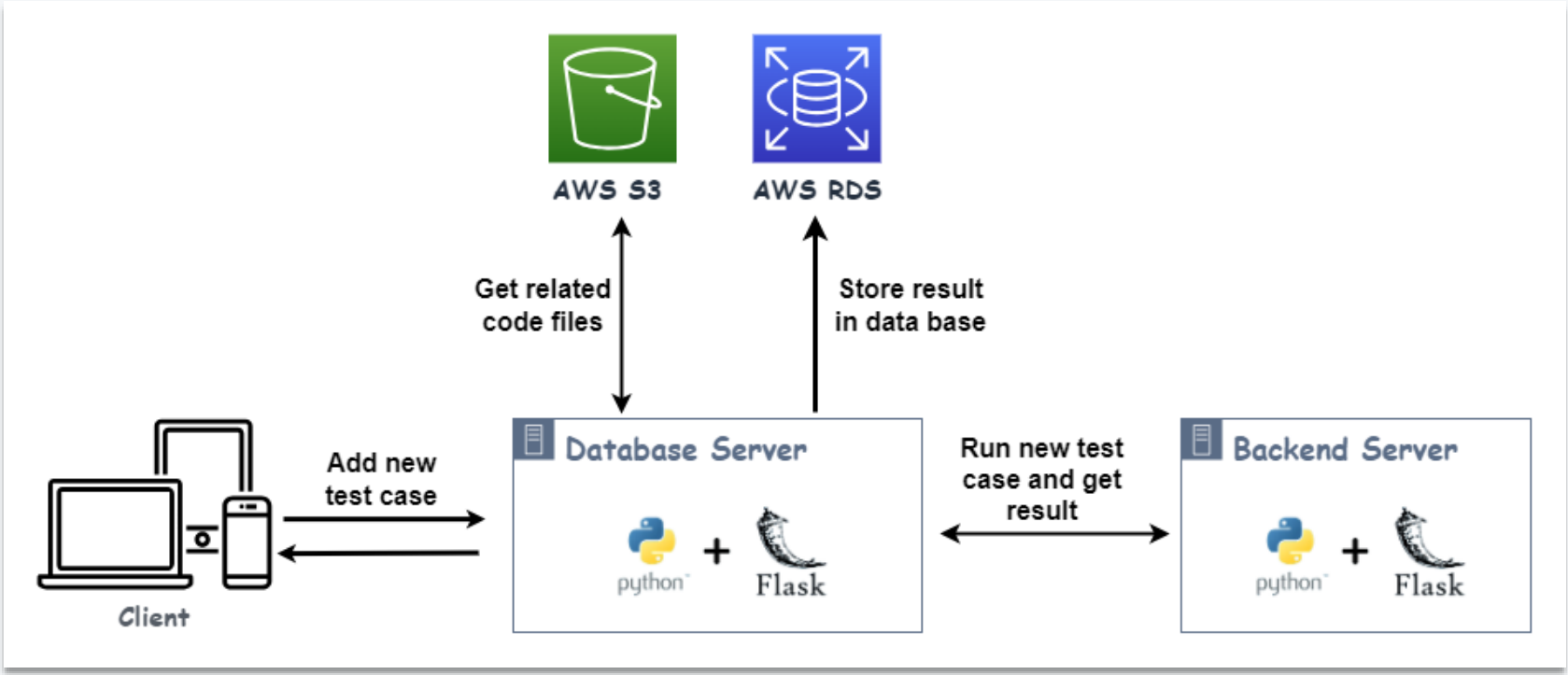


Professor Features (cont.)

Add new test case when there is a submission for challenge

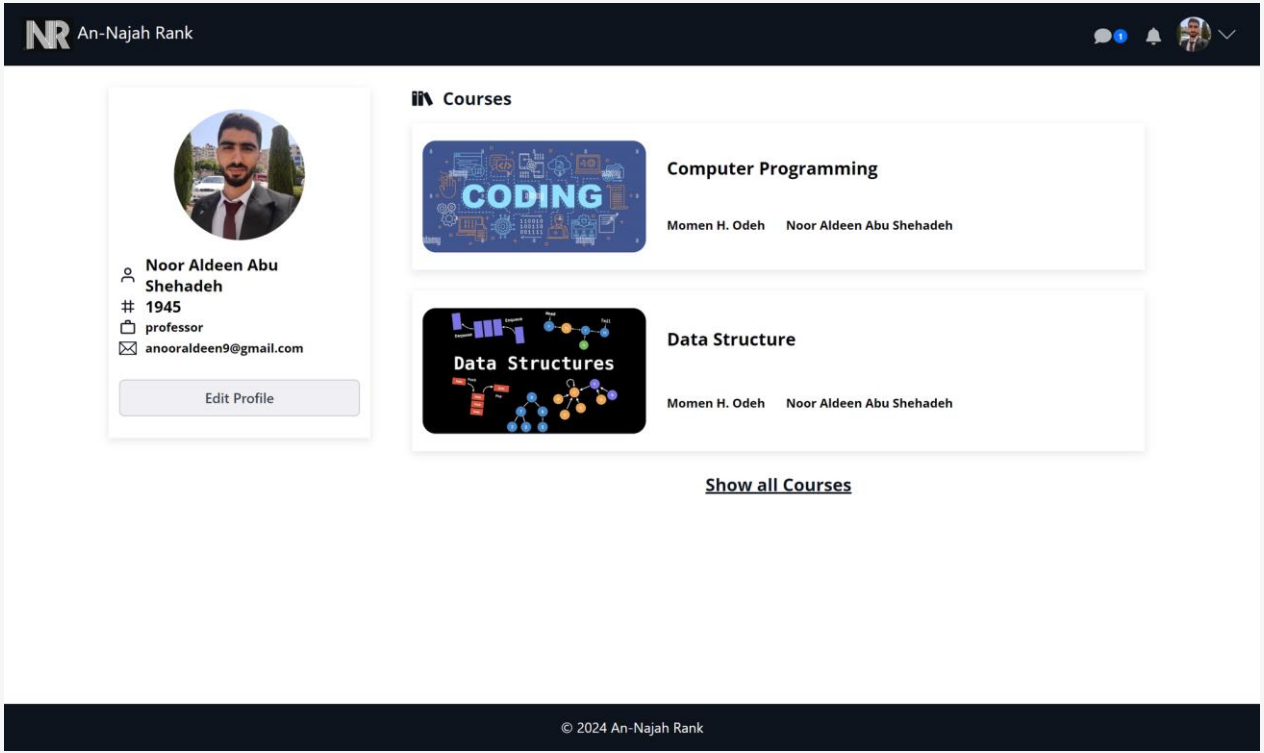


Add new test case when there is a submission for challenge (cont.)



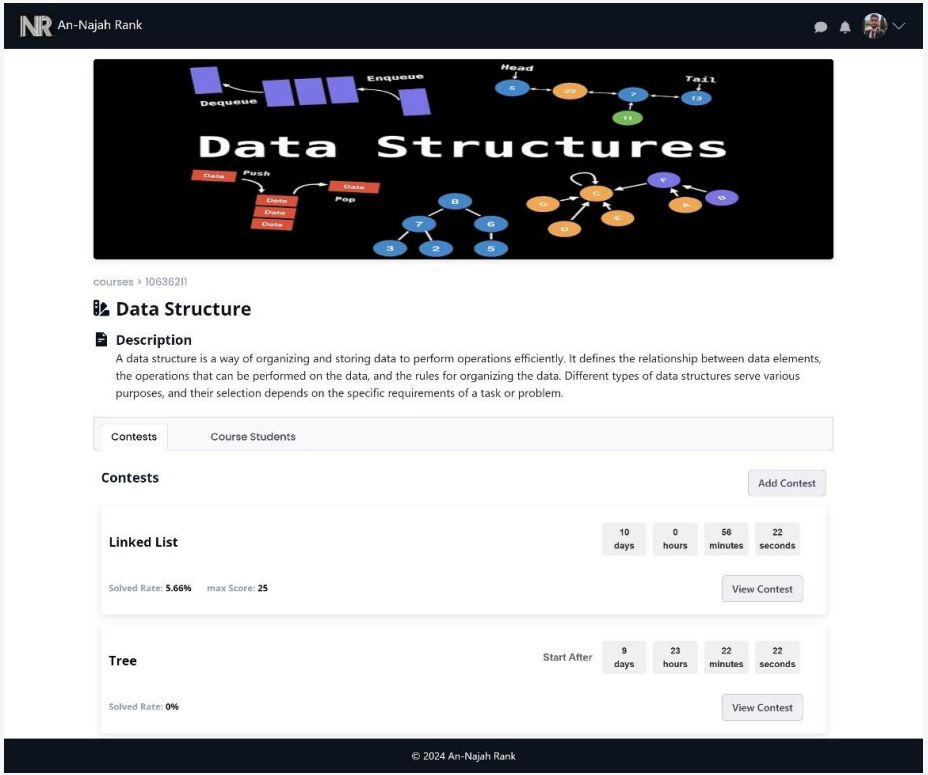
Professor Features (cont.)

Profile:



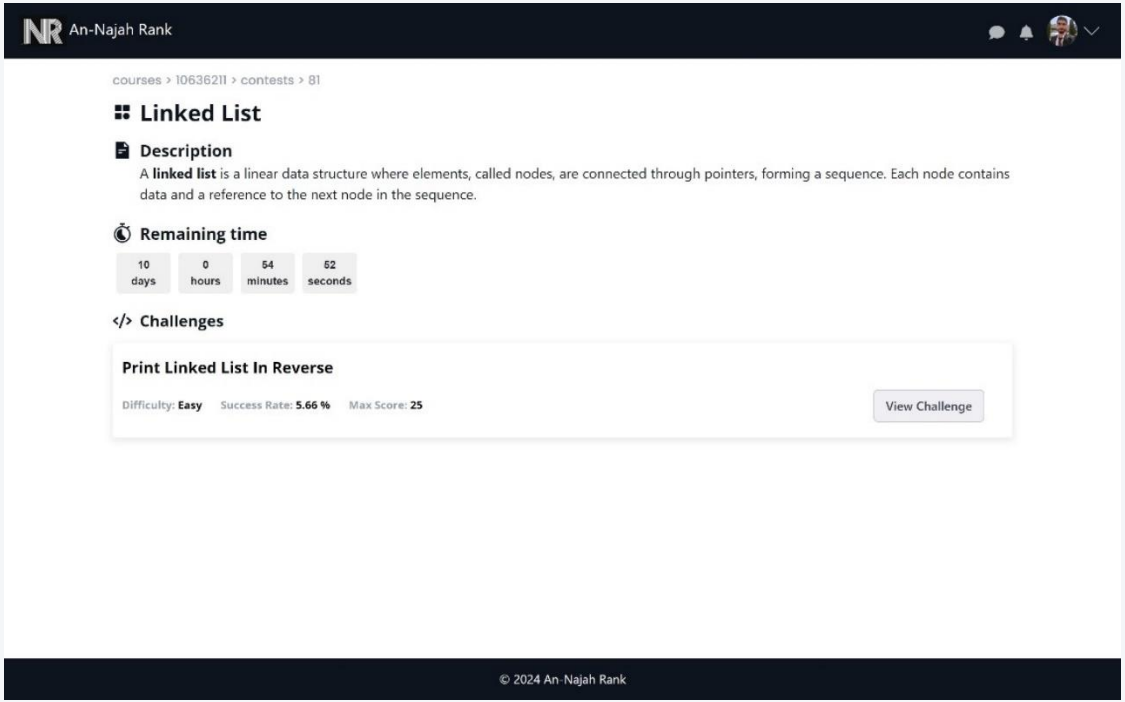
Professor Features (cont.)

Course View:



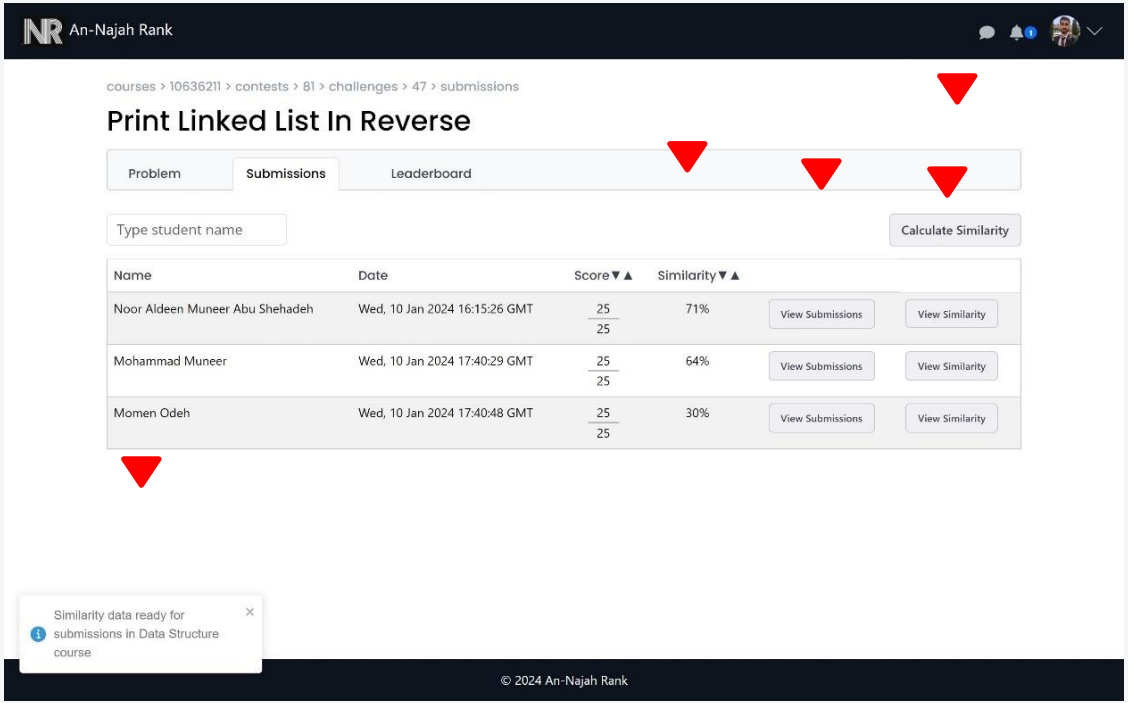
Professor Features (cont.)

Contest View:



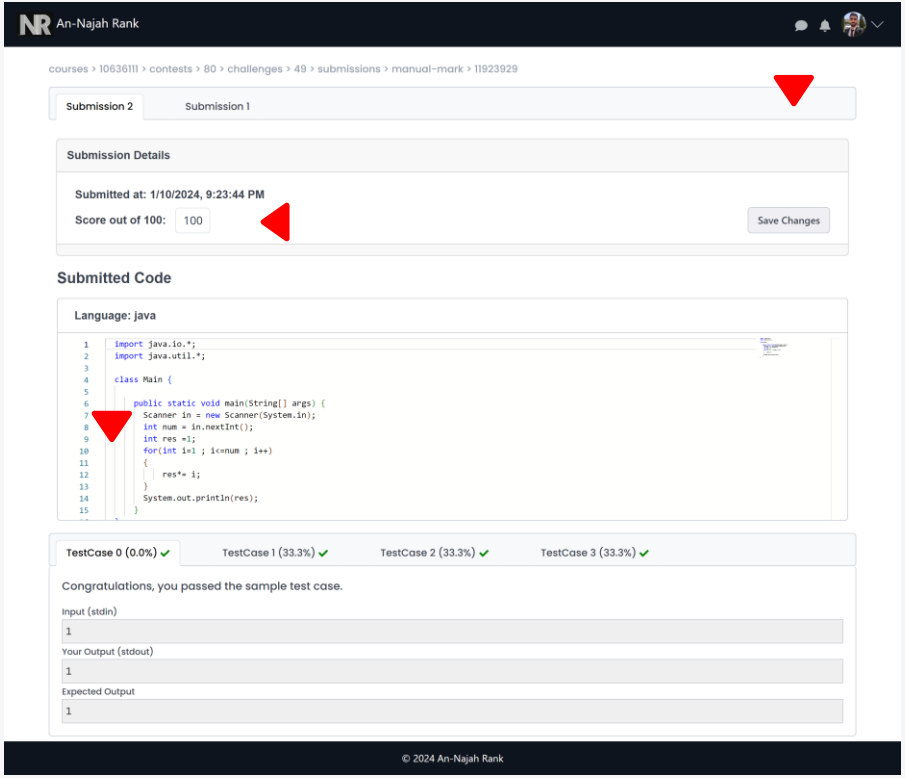
Professor Features (cont.)

Challenge submission View:



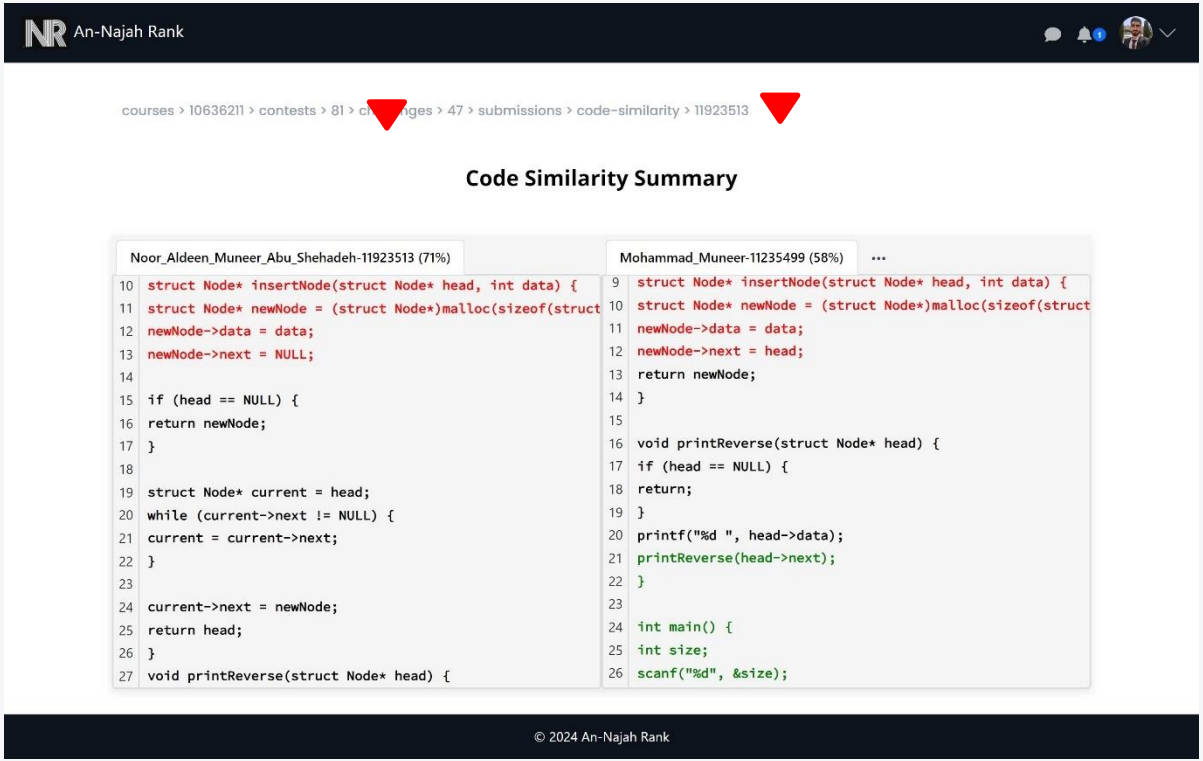
Professor Features (cont.)

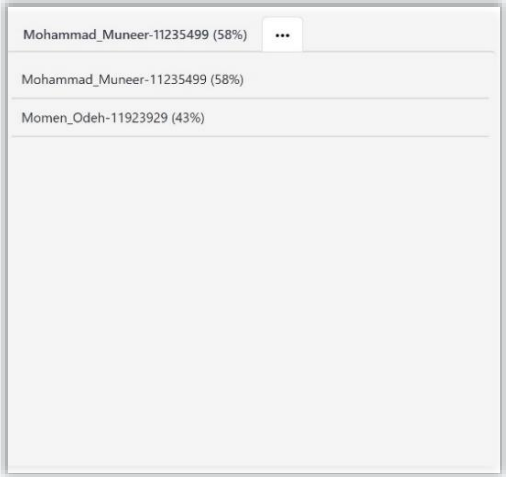
Challenge student submissions View:



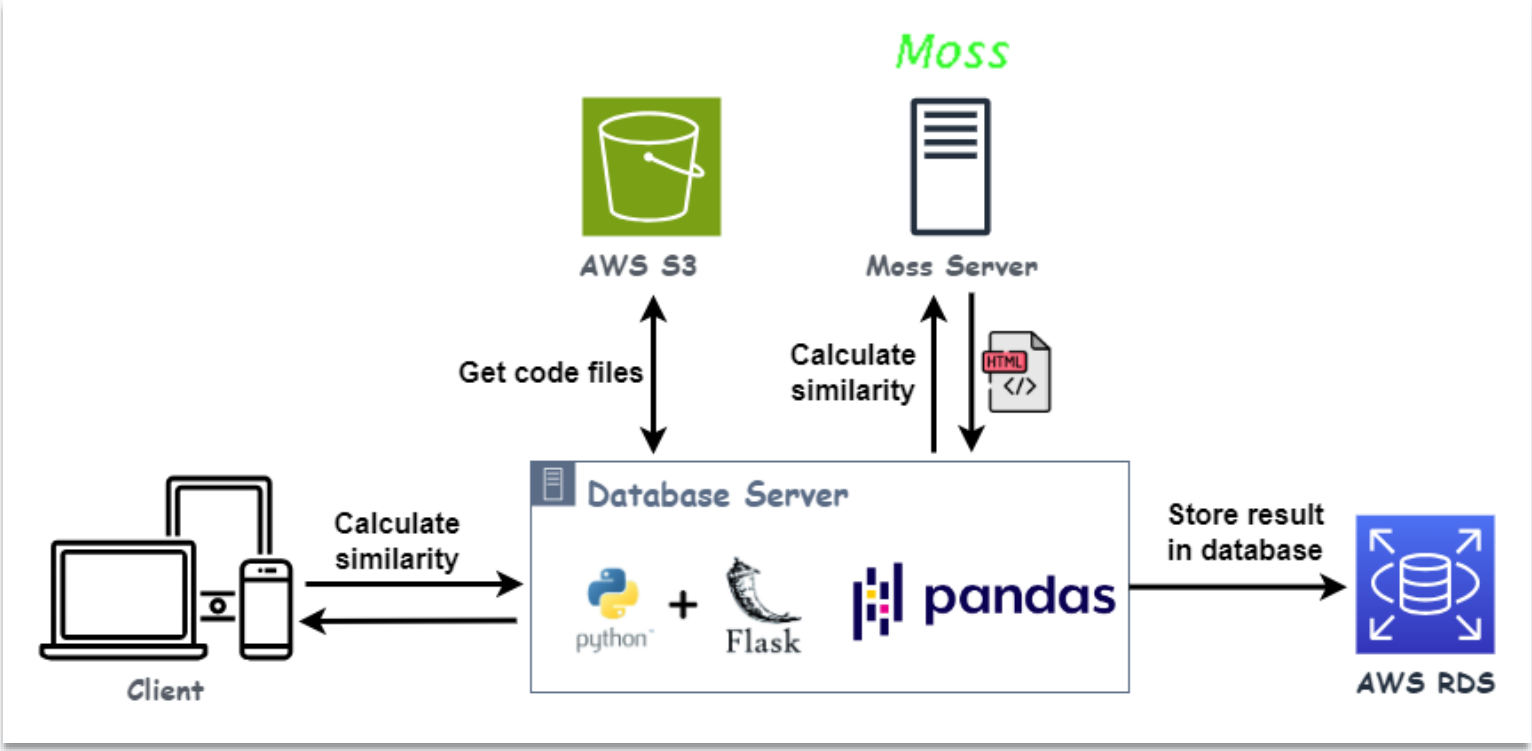
Professor Features (cont.)

Challenge student similarity View:



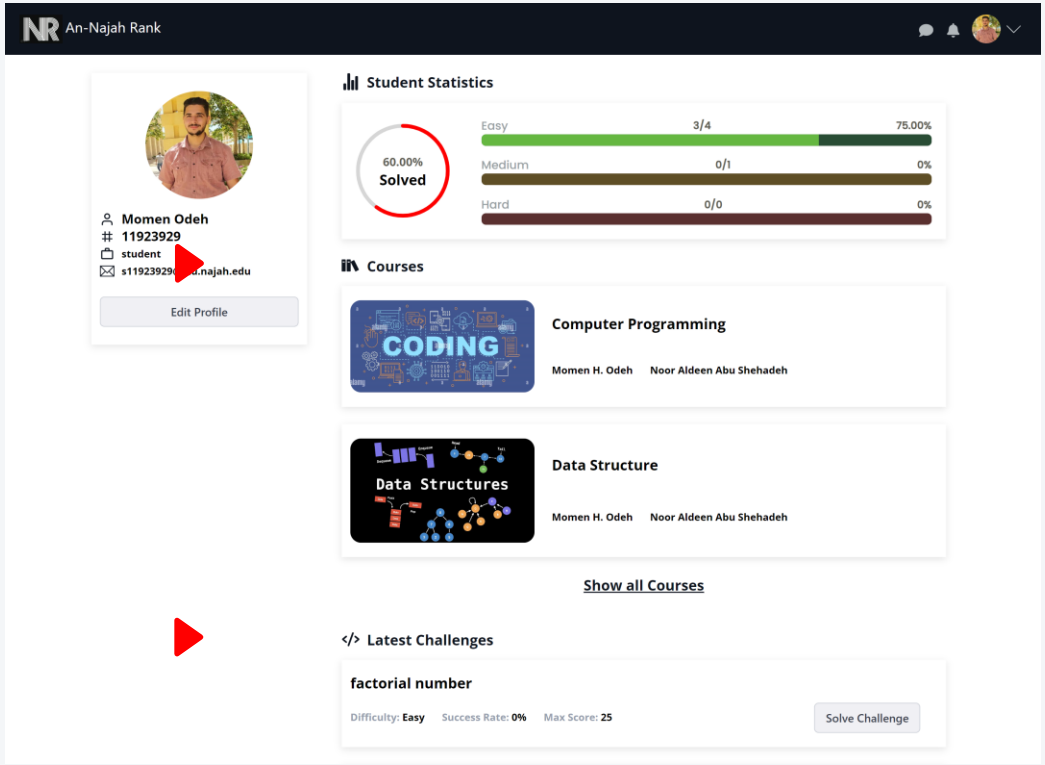


Challenge student similarity View (cont.):



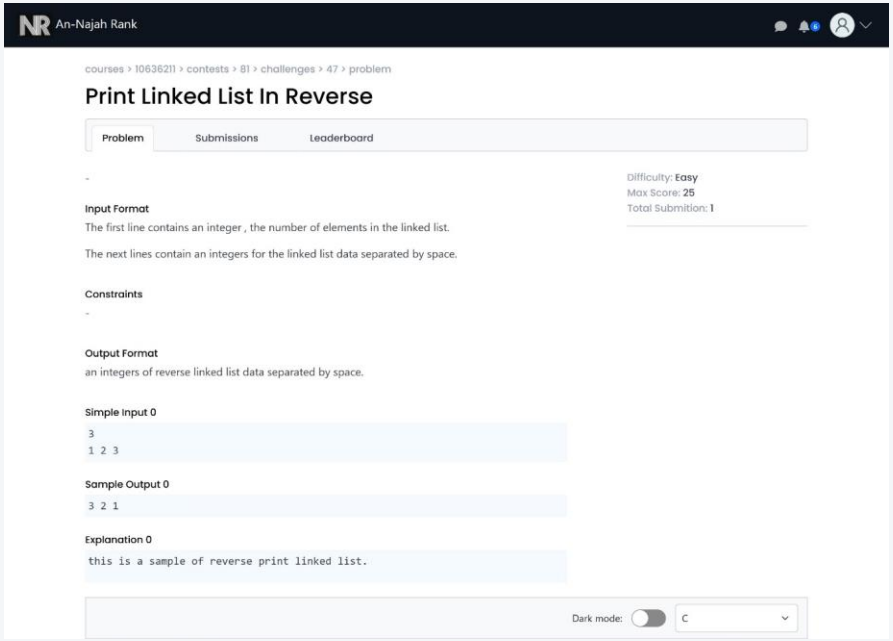
Student Features

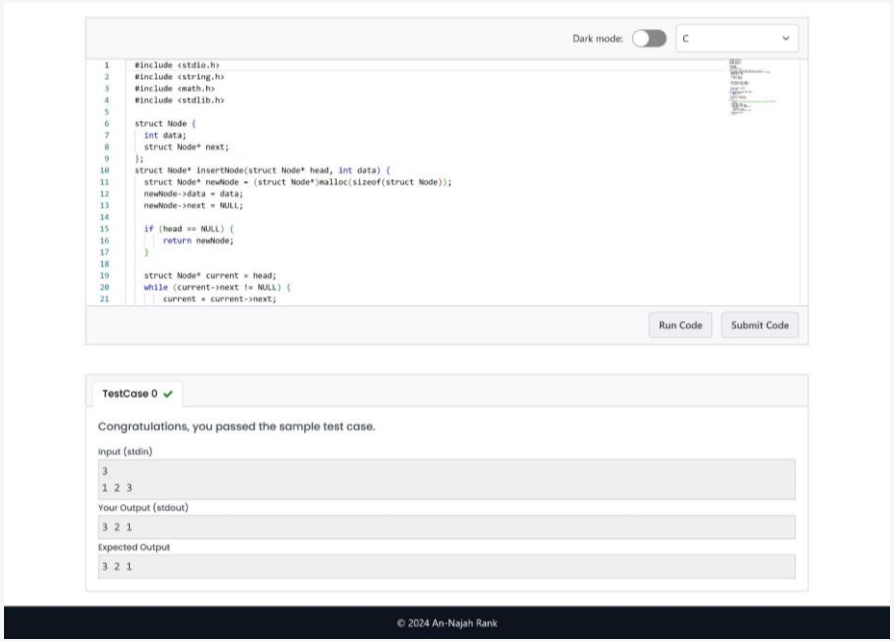
Student Profile:



Student Features

Solving challenge:



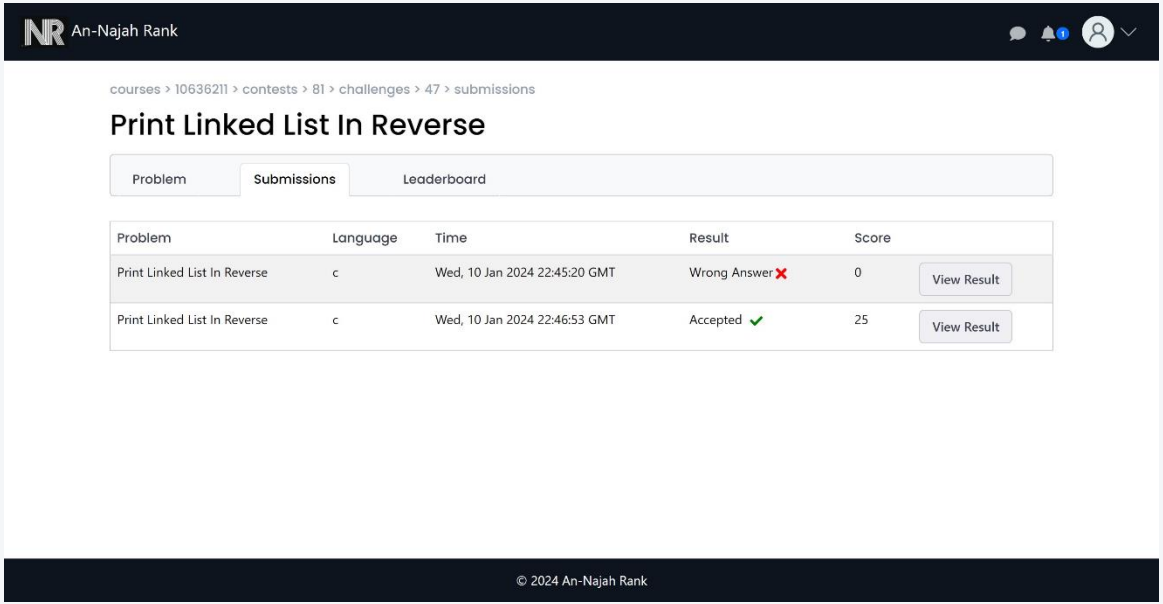


Code operation:

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

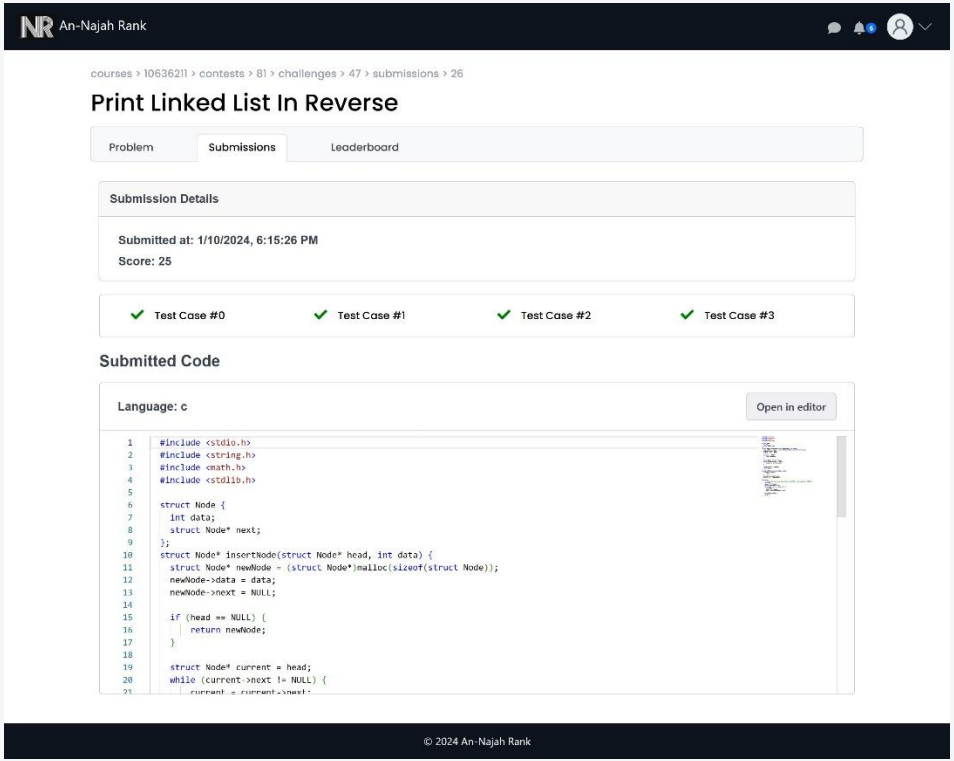
Student Features

Student submission:

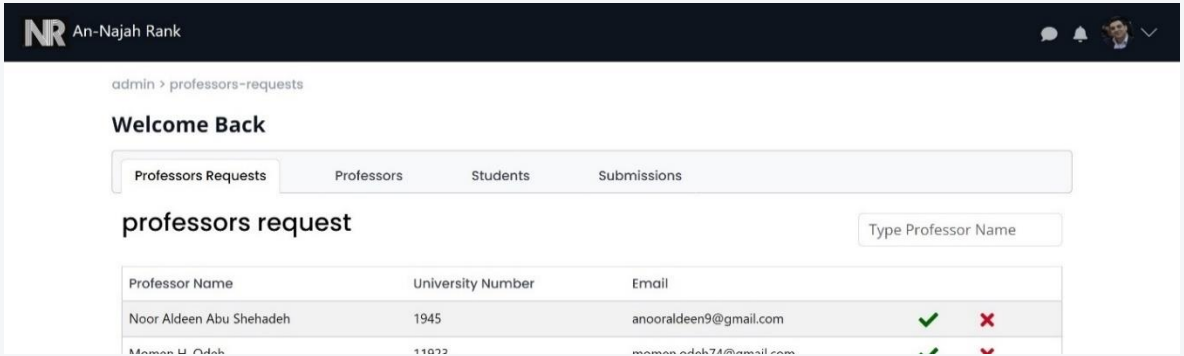


Student Features

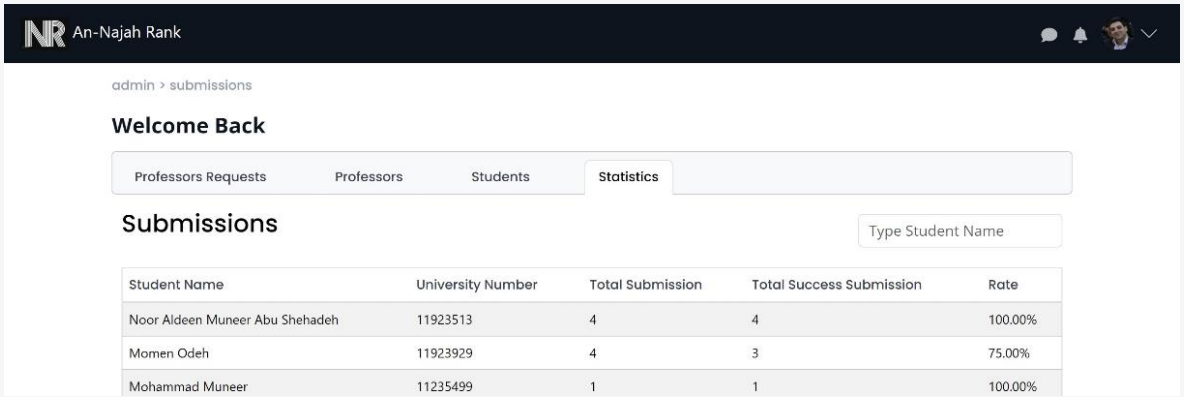
Student submission (cont.):



Admin Features

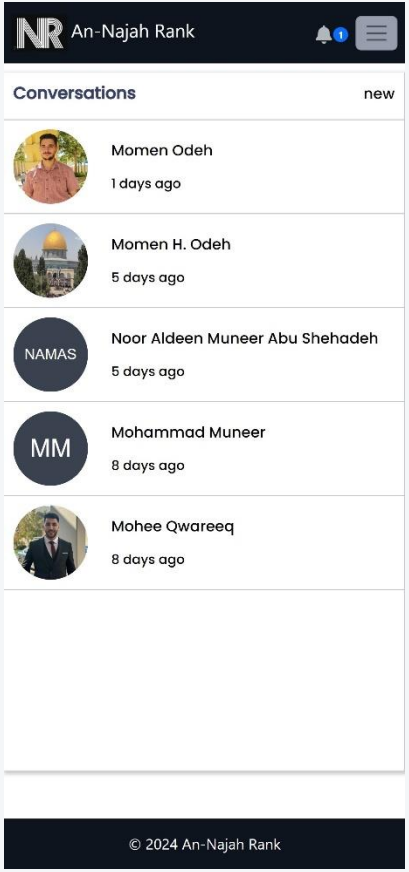


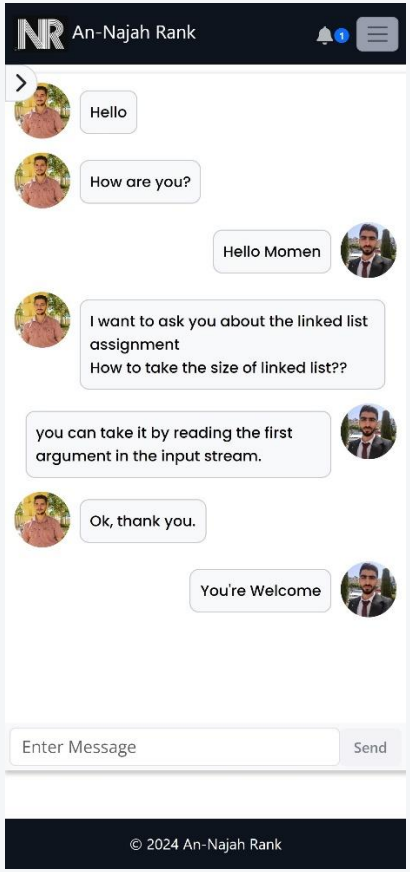
pending professors

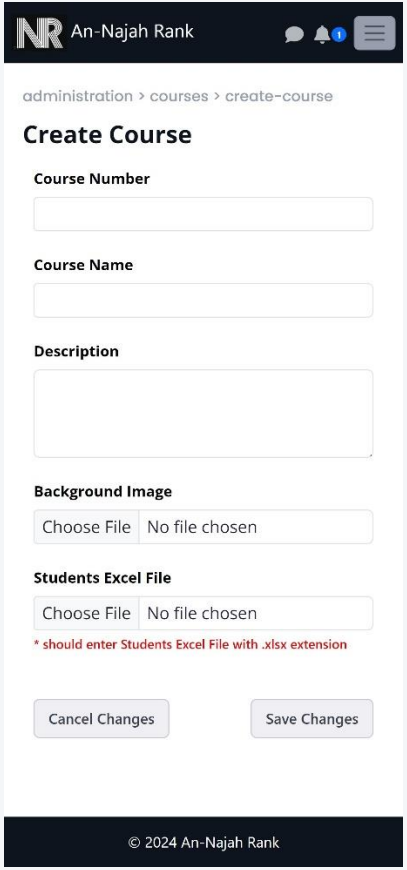


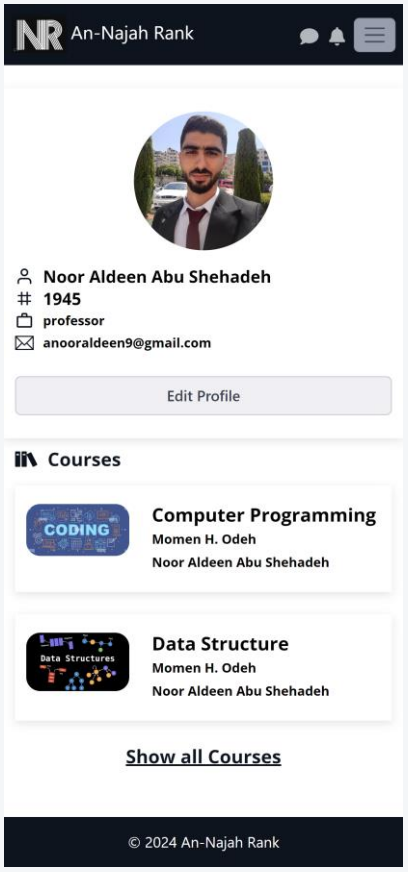
Students statistics

Sample of responsive design

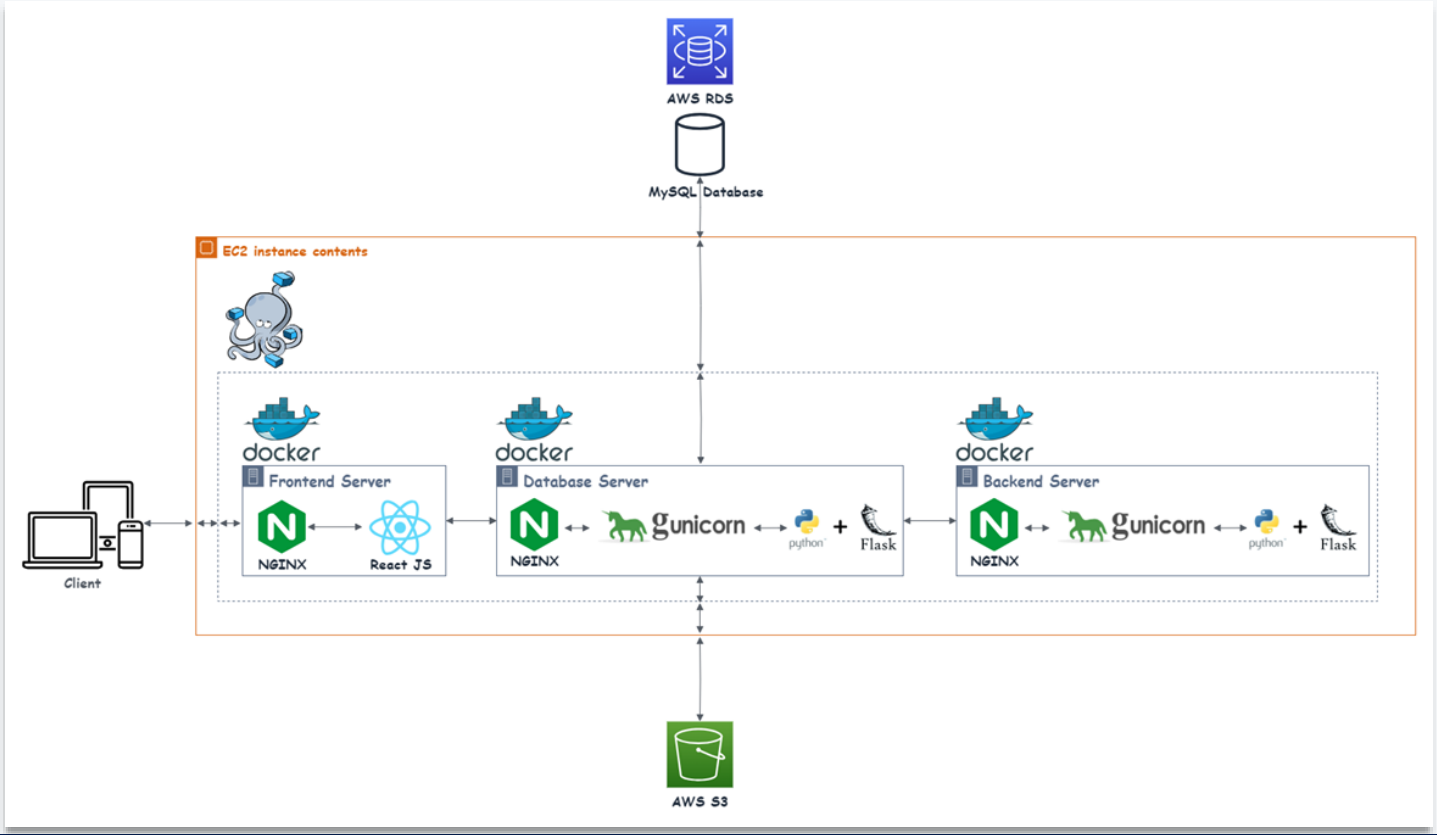








Deployment Phase

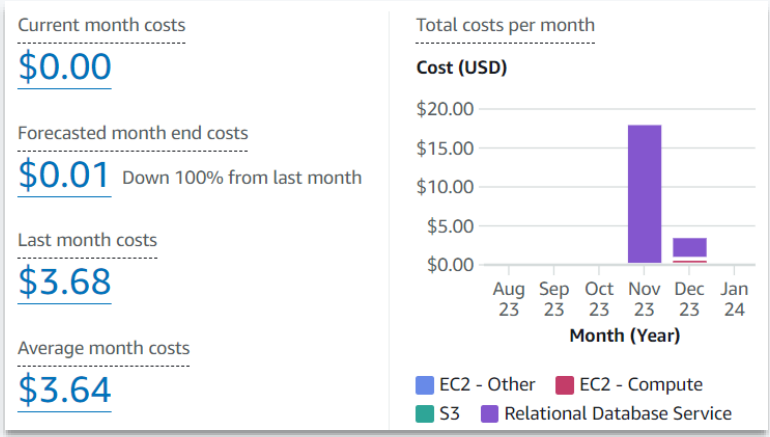


Testing Phase

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 !