



# CS 353 FINAL REPORT

Online Coding Platform

GROUP 14

Ahmet Hakan Yılmaz 21803399

Akın Kutlu 21803504

Selbi Ereshova 21901326

Zehra Erdem 21801977

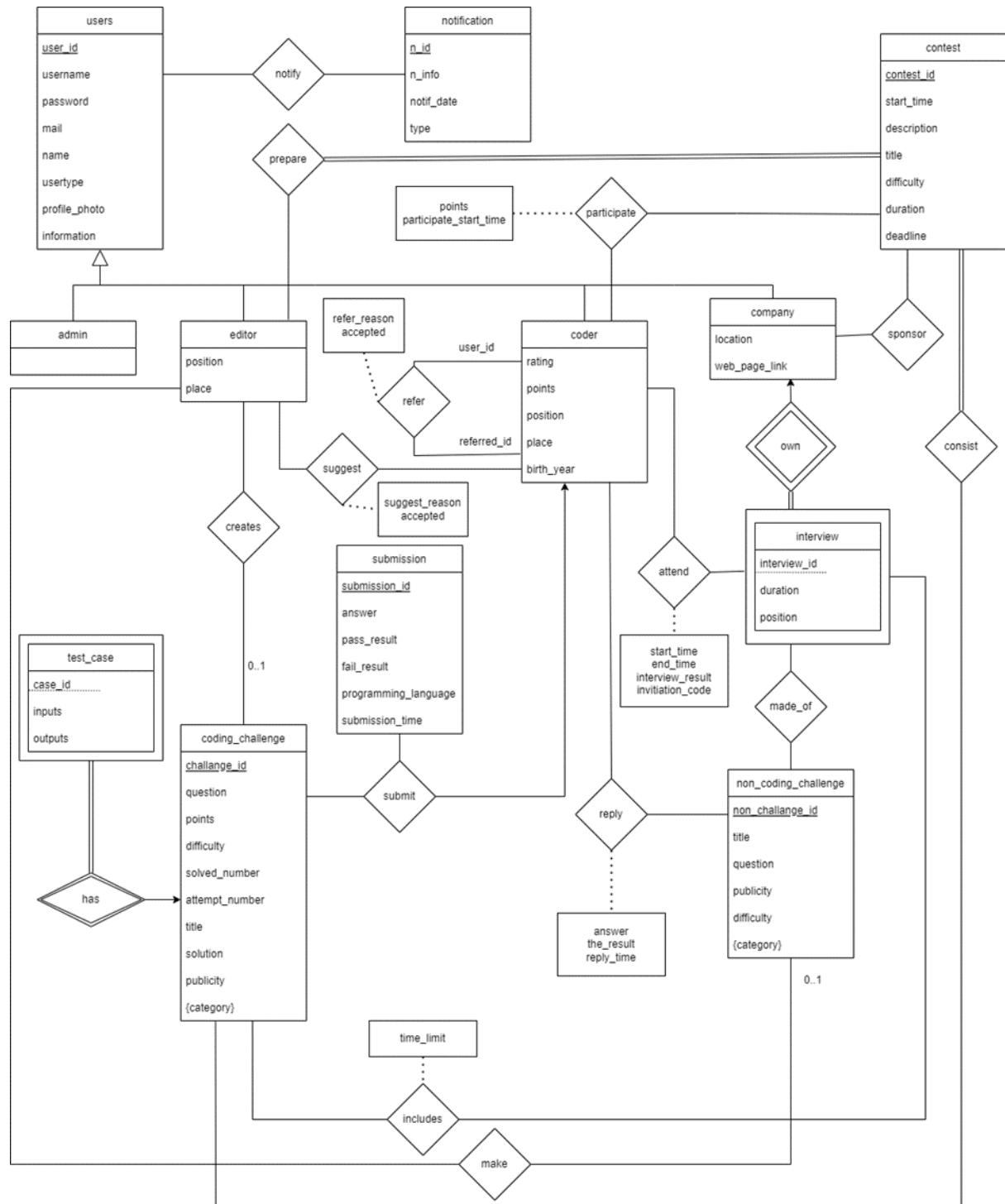
<b>Description</b>	<b>3</b>
<b>ER diagram</b>	<b>4</b>
<b>List Of Tables</b>	<b>5</b>
<b>Implementation Details</b>	<b>15</b>
<b>Advanced DB features</b>	<b>16</b>
Reports	16
Views	18
Constraints	18
Indexes	19
<b>Contributions</b>	<b>20</b>
<b>User Manual</b>	<b>21</b>
<b>General Functions</b>	<b>21</b>
1. Login	21
2. List of Coding Challenges	21
2.1. Filtering for Coding Challenges	22
3. List of Non-Coding Questions	22
3.1. Filtering for Non-Coding Questions	23
4. Details of A Contest	24
5. List of Coders	25
5.1. Search for Coders	25
<b>Coder</b>	<b>26</b>
1. Registration	26
2. Menu	26
3. Solving Coding Challenge	27
3.1 See Old Submissions	27
3.2 See Details of Coding Challenge	28
4. Solving Non-Coding Question	28
4.1. See Your Submission	29
4.2 See Details of Non Coding Question	29
4.3 See Others' Answer	30
5. Contests	30
5.1 Future and Non Registered Contests	30
5.2 List of Future and Registered Contests	31
5.3 List of Started Contests	32
5.4. List of Finished Contests	34
6. See Other Coders's Profile	35
6.1. Give Referral to Another Coder	35
7. See Your own Profile	36
8. See Notifications	36
9. See Interviews	37

9.1 See Old Interviews	37
9.2 See New Interviews	38
9.3 Filter New Interviews	38
10. Attend Interview	39
10.1 Start Interview	39
10.2 Attend Interview	39
<b>Editor</b>	<b>40</b>
1.Register	40
2. Menu	40
3. See Coding Challenge	41
4. See Non Coding Question	41
5. Create Coding Challenge	42
6. Create Non-Coding Question	42
7. Create Contest	43
7.1. Add Challenges to Contest	43
8. Contests	44
8.1 List of AllContests	44
8.2. List of Your Contests	44
9. See Coder's Profile	45
9.1 Give Referral to A Coder	45
<b>Company</b>	<b>46</b>
1. Register	46
2. Menu	46
3. See Coding Challenge	47
4. See Non Coding Question	47
5. Contests	48
5.1. List of All Contests	48
5.2. Sponsored Contests	48
6. Be Sponsor to A Contest	49
7. See Profile of A Coder	49
8. Create Interview	50
8.1 Create Interview	50
8.2 Add Coding Questions to Interview	50
8.3 Add Non-Coding Questions to Interview	51
8.4. Submit Interview	52
8.4 Add Participants to Interview	53
9. See and Evaluate Interviews	53
9.1 See the List of Interview	53
9.2 See the List of Candidates	54
9.3 See Candidate's Results	54
9.4 Evaluate Candidate	55

# Description

We developed a coding platform web application. Three types of users can benefit from this application. Coder users can view and solve coding and non-coding questions. Coders can view the information of the coding and non-coding questions and also view their previous submissions if they have any. While solving coding questions, coders can specify their choice of programming language. For non-coding questions, coders can view other coders' submissions after they have made their own submission. Coders can also participate in contests and attend interviews. For each contest, users can see the leaderboard. Each coder has a rating and point. They can increase their points by correctly solving non-coding and coding questions in contests or outside of contests. Ratings of coders are calculated according to the average success of their in contests. Company users can also view coding and non-coding questions. A company can sponsor contests, create interviews and evaluate the candidates for the interview. While creating an interview, a company can add new coding and non-coding questions to the interview. They can also specify which users to add as candidates and give deadlines for each. Editor user types can view and create coding and non-coding questions. They can also create contests consisting of their created coding and non-coding questions.

# ER diagram



# List Of Tables

In this part, relational models, primary keys and foreign keys of tables are given.

## 2.1 users

### **Relational Model**

users(user\_id, username, password, mail, name, usertype, profile\_photo, information)

### **Primary Key**

user\_id

### **Foreign Keys**

None

## 2.2 admin

### **Relational Model**

admin(user\_id)

### **Primary key**

user\_id

### **Foreign Keys**

user\_id references users(user\_id)

## 2.3 editor

### **Relational Model**

editor(user\_id, position, place)

### **Primary key**

user\_id

### **Foreign Keys**

user\_id references users(user\_id)

## 2.4 company

### **Relational Model**

company(user\_id, location, web\_page\_link)

#### **Primary key**

user\_id

#### **Foreign Keys**

user\_id references users (user\_id)

## 2.5 coder

### **Relational Model**

coder(user\_id, rating, points, position, place, birth\_year)

#### **Primary key**

user\_id

#### **Foreign Keys**

user\_id references users (user\_id)

## 2.6 refer

### **Relational Model**

refer(user\_id, referred\_id, refer\_reason, accepted )

#### **Primary Key**

user\_id, referred\_id

#### **Foreign Keys**

user\_id references coder(user\_id)

referred\_id references coder(user\_id)

## 2.7 suggest

### **Relational Model**

suggest(user\_id, coder\_id, suggest\_reason, accepted)

### **Primary Key**

user\_id, coder\_id

### **Foreign Keys**

user\_id references editor(user\_id)

coder\_id references coder(user\_id)

## 2.8 contest

### **Relational Model**

contest(contest\_id, start\_time, description, title, difficulty, duration, deadline)

### **Primary key**

contest\_id

### **Foreign Keys**

None

## 2.9 participate

### **Relational Model**

participate(contest\_id, user\_id, points, participate\_start\_time )

### **Primary key**

contest\_id, user\_id

### **Foreign Keys**

contest\_id references contest(contest\_id)

user\_id references coder(user\_id)

## 2.10 sponsor

### **Relational Model**

sponsor(contest\_id, user\_id)

#### **Primary keys**

contest\_id, user\_id

#### **Foreign Keys**

contest\_id references contest(contest\_id)

user\_id references company(user\_id)

## 2.11 prepare

### **Relational Model**

prepare(contest\_id, user\_id)

#### **Primary keys**

contest\_id, user\_id

#### **Foreign Keys**

contest\_id references contest(contest\_id)

user\_id references editor(user\_id)

## 2.12 notification

### **Relational Model**

notification(n\_id, n\_info, notif\_date, type)

#### **Primary keys**

n\_id

#### **Foreign Keys**

None

## 2.13 notify

### **Relational Model**

notify(user\_id, n\_id)

### **Primary keys**

user\_id, n\_id

### **Foreign Keys**

user\_id references users (user\_id)

n\_id references notification(n\_id)

## 2.14 coding\_challenge

### **Relational Model**

coding\_challenge(challenge\_id, question, points, difficulty, solved\_number, attempt\_number, title, solution, publicity)

### **Primary keys**

challenge\_id

### **Foreign Key**

-

## 2.15 coding\_challenge\_categories

### **Relational Model**

coding\_challenge\_categories(challenge\_id, category)

### **Primary Key**

challenge\_id, category

### **Foreign Key**

challenge\_id references coding\_challenge(challenge\_id)

## 2.16 non\_coding\_challenge

### **Relational Model**

non\_coding\_challenge(non\_challenge\_id, question, difficulty, title, publicity)

#### **Primary Key**

non\_challenge\_id

#### **Foreign Key**

-

## 2.17 non\_coding\_challenge\_categories

### **Relational Model**

non\_coding\_challenge\_categories(non\_challenge\_id, category)

#### **Primary Key**

non\_challenge\_id, category

#### **Foreign Key**

non\_challenge\_id references non\_coding\_challenge(non\_challenge\_id)

## 2.18 submission

### **Relational Model**

submission(submission\_id, answer, pass\_result, fail\_result, programming\_language, , submission\_time)

#### **Primary Key**

submission\_id

#### **Foreign Key**

-

## 2.19 submit

### **Relational Model**

submit(challenge\_id, submission\_id, user\_id)

### **Primary Key**

challenge\_id, submission\_id, user\_id

### **Foreign Key**

challenge\_id references coding\_challenge(challenge\_id)

submission\_id references submission(submission\_id)

user\_id references coder(user\_id)

## 2.20 test\_case

### **Relational Model**

test\_case(challenge\_id, case\_id, inputs, outputs)

### **Primary Key**

challenge\_id, case\_id

### **Foreign Key**

challenge\_id references coding\_challenge(challenge\_id)

## 2.21 creates

### **Relational Model**

creates (user\_id, challenge\_id)

### **Primary Key**

user\_id, challenge\_id

### **Foreign Key**

user\_id references editor(user\_id)

challenge\_id references coding\_challenge(challenge\_id)

## 2.22 interview

### **Relational Model**

interview (user\_id,interview\_id, duration,position)

#### **Primary Key**

user\_id,interview\_id

#### **Foreign Key**

user\_id references company(user\_id)

## 2.23 includes

### **Relational Model**

includes (challenge\_id,interview\_id,company\_id, time\_limit)

#### **Primary Key**

challenge\_id, interview\_id,company\_id

#### **Foreign Key**

challenge\_id references coding\_challenge(challenge\_id)

interview\_id references interview(interview\_id)

company\_id references interview(user\_id)

## 2.24 consist

### **Relational Model**

consist (challenge\_id,contest\_id)

#### **Primary Key**

challenge\_id, contest\_id

#### **Foreign Key**

challenge\_id references coding\_challenge(challenge\_id)

contest\_id references contest(contest\_id)

## 2.25 reply

### **Relational Model**

reply (non\_challenge\_id,user\_id,answer,the\_result, reply\_time)

### **Primary Key**

non\_challenge\_id, user\_id

### **Foreign Key**

non\_challenge\_id references non\_coding\_challenge(non\_challenge\_id)

user\_id references coder(user\_id)

## 2.26 attend

### **Relational Model**

attend (interview\_id,coder\_id,company\_id, start\_time,end\_time,interview\_result,invitation\_code)

### **Primary Key**

interview\_id,coder\_id,company\_id

### **Foreign Key**

interview\_id references interview(interview\_id)

coder\_id references coder(user\_id)

company\_id references interview(user\_id)

## 2.27 make

### **Relational Model**

make (non\_challenge\_id,user\_id)

### **Primary Key**

non\_challenge\_id,user\_id

### **Foreign Key**

non\_challenge\_id references non\_coding\_challenge(non\_challenge\_id)

user\_id references editor(user\_id)

## 2.28 made\_of

### **Relational Model**

made\_of (non\_challenge\_id,interview\_id,user\_id)

### **Primary Key**

non\_challenge\_id,interview\_id,user\_id

### **Foreign Key**

non\_challenge\_id references non\_coding\_challenge(non\_challenge\_id)

interview\_id references interview(interview\_id)

user\_id references interview(user\_id)

# Implementation Details

For backend implementation, we used Spring Boot Framework, Java Language and Maven environment because we were familiar with them. For the frontend side, we used React.js. For SQL, we used PostgreSQL and AWS as our service.

For connection between SQL and Java, we created a ConnectionSingle object with singleton principle. We connected to our SQL server by using the code that is in constructor of ConnectionSingle :

```
try {
    connection = DriverManager.getConnection(url, user, password);
} catch (SQLException throwables) {
    throwables.printStackTrace();
}
```

To run a query, we used PreparedStatement which is a part of the java.sql library. With the set function, we can set the necessary pass parameters to sql. After executing the query (executeQuery), we used next() and get() functions to get data from the result of the query.

A simple query example:

```
try {
    String getChallengeSql = "Select * From test_case where challenge_id = ?";
    PreparedStatement insertCodingPrepared = ConnectionSingle.getConnection().prepareStatement(getChallengeSql);
    insertCodingPrepared.setInt(1, challenge_id);
    ResultSet rs = insertCodingPrepared.executeQuery();
    while (rs.next()) {
        result.add(rs.getString("outputs"));
    }
} catch (SQLException throwables) {
    throwables.printStackTrace();
}
```

To show clear error and success messages, we create and send MessageResponse objects which return messageType and message.

We encountered several problems. Saving and reading date and time, we used Postgresql TIMESTAMP data type. In Java, there is a timestamp so this data type can easily be used. However, in some queries, we needed to compare a timestamp with current time and it gives the wrong result (start\_time > CURRENT\_TIMESTAMP). The reason for it was different time zones. We solved the problem by adding +3 hours when comparing (start\_time > CURRENT\_TIMESTAMP + INTERVAL '03:00') because the time difference between UTC and Turkey is 3 hours. Another problem was that some queries were too long to write and complicated to understand when looked at later. Instead of long queries, we created views and used them.

# Advanced DB features

## Reports

(I) To report how many coding challenges have been created for a category.  
Similar one used for non coding questions as well.

SQL:

```
Select *
From (SELECT Count(*) as cat_number, category
      FROM coding_challenge_categories
      group by category order by cat_number desc ) sub
```

Output got by executing the SQL:

Data Output		Explain	Messages	Notifications
	cat_number bigint	category character varying (31)		
1	11	DP		
2	10	Graph		
3	5	Array		
4	4	BFS		
5	4	DFS		
6	3	Linked list		
7	3	Stack		

Display in Screen of Our Application

Page: List of Coding Challenge

Welcome LOGOUT

Filter for Coding Challenges:

Categories CLEAR

DP : 11 Graph : 10 Array : 5 BFS : 4 DFS : 4 Linked list : 3 Stack : 3

Merge k Sorted Lists Difficulty: Hard Point: 90 Number of Attempts: 11 Number of Accepted Solutions: 4 <a href="#">SEE CHALLENGE</a>	Coin Change Difficulty: Medium Point: 80 Number of Attempts: 5 Number of Accepted Solutions: 3 <a href="#">SEE CHALLENGE</a>	Change Difficulty: Medium Point: 80 Number of Attempts: 1 Number of Accepted Solutions: 1 <a href="#">SEE CHALLENGE</a>
---	---	--

(II) To get reports about a contest. In this query average, max and min points are calculated as well as number of participants

## SQL:

```
SELECT *
FROM (SELECT contest_id, count(*) as userNumber, max(points) as maxPoint, avg(points)
as avgPoint, min(points) as minPoint
      FROM participate group by contest_id) sub
```

### Output got by executing the SQL:

### Display in Screen of Our Application

Page: Details of A Contest

Author	Number of Participants
Hakan	8
Description	Average Point
weekly Contest 16	142
Difficulty	Max Point
70	295
Start Time	Min Point
16-05-2022 12:12	0
Deadline	
17-05-2022 14:12	

## Views

(I) Users and coder tables are used so often that a join table is added to ease this process and for privacy and security, user password is not included.

SQL:

```
CREATE VIEW userCodersView AS SELECT U.user_id, U.username, U.mail, U.name, U.profile_photo, U.information, C.rating, C.position, C.points, C.place, C.birth_year FROM coder C, users U where U.user_id = C.user_id
```

Example Used SQL:

```
SELECT * from userCodersview WHERE username LIKE ? order by rating DESC
```

Display in Screen of Our Application

Data Output Explain Messages Notifications											
	user_id	username	mail	name	profile_ph	information	rating	position	points	place	birth_year
1	8	Sabri	sabri@gm...	Sabri	[null]	[null]	3144	Intern	235	JPMorgan	1999
2	7	Sude	sude_ari...	Sude Ari	[null]	[null]	2337	Intern	436	KU	2000
3	6	Sena	senag@g...	Sena Genc	[null]	[null]	540	Student	41	Bilkent University	2001
4	2	SelbiCoder	selbi@gm...	Selbi	[null]	[null]	0	Student	0	Bilkent	2000

Display in Screen of Our Application:

Page: Coders

The screenshot shows a web application interface. At the top, there is a blue header bar with the text "Welcome" and a "LOGOUT" button. Below the header, there is a search bar with a placeholder "Username" and a search icon. To the right of the search bar is a "CLEAR" button. Below the search bar is a table with the following columns: Placement, Username, Name, Place, Position, Rating Point, and Profile. The table contains four rows of data corresponding to the users listed in the previous table.

Placement	Username	Name	Place	Position	Rating Point	Profile
1	Sabri	Sabri	JPMorgan	Intern	3144	SEE PROFILE
2	Sude	Sude Ari	KU	Intern	2337	SEE PROFILE
3	Sena	Sena Genc	Bilkent University	Student	540	SEE PROFILE
4	SelbiCoder	Selbi	Bilkent	Student	0	SEE PROFILE

## Constraints

We also have 2 more constraints that check the times are appropriate. One is in the contest table and other is in the attend table both are

In contest table we have the constraint:

```
CHECK ( deadline > start_time)
```

In attend table we have the constraint:

```
CHECK ( end_time > start_time)
```

## Indexes

(I) coderPointIndex is added because coders are sorted according to their points and using indexes make sorting fast.

SQL:

```
CREATE INDEX coderPointIndex ON coder(points)
```

(II) coderRatingIndex is added because coders are sorted according to their ratings and using indexes make sorting fast.

SQL:

```
CREATE INDEX coderRatingIndex ON coder(rating)
```

(III) usersUsernameIndex is added because users are found according to their username and using indexes make searching fast.

SQL:

```
CREATE INDEX usersUsernameIndex ON users(username)
```

# Contributions

**Ahmet Hakan Yilmaz:** I worked mainly on sql and backend of the project. I worked mainly on database components and some parts of the backend like non coding challenges, submissions, interviews and contests .

**Akin Kutlu:** I worked on the backend of the project. Specifically, I worked on developing login, registration for different types of users, coding challenges, notifications and references for coders and editors.

**Selbi Ereshova:** I worked on the frontend of the project. Specifically, I worked on creating coding, non coding questions, profile pages, giving referral to coders, creating and evaluating interviews as a company. Seeing interviews as a coder and attending them.

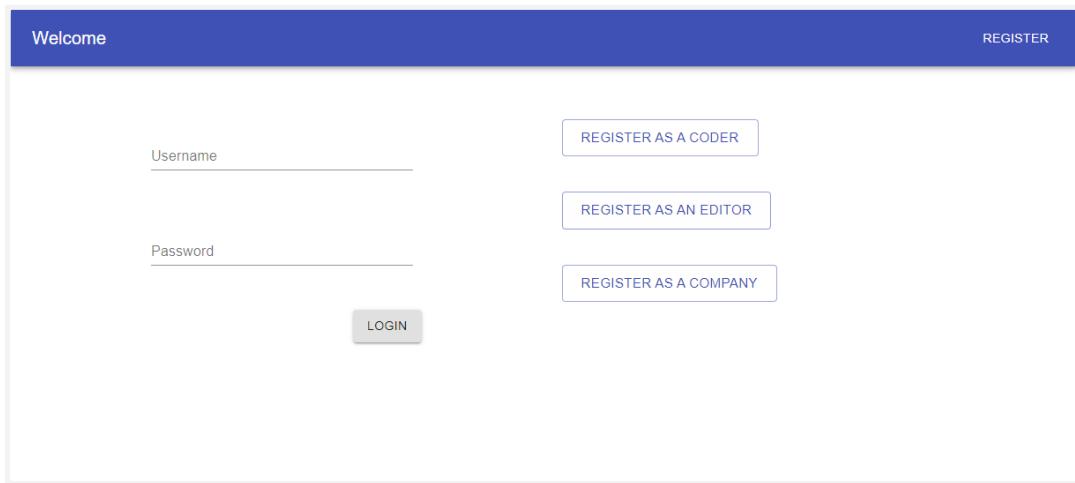
**Zehra Erdem:** I worked on the frontend of the project. I worked on creating coding, non coding questions as well. I worked on creating, registering and solving contests, seeing contest details which includes the leaderboard of the contest. I also work on the coders table and notification parts.

# User Manual

## General Functions

### 1. Login

Every user type can login from this page. For new users, registration page also can be reachable from here.



### 2. List of Coding Challenges

Every user can see a list of coding challenges by clicking “Coding Challenges” in the menu. By pressing the “See Challenge” button, different types of details will be open for different types of users.

A screenshot of the challenge list page. At the top left is a 'Welcome' message and at the top right are 'LOGOUT' and profile icons. Below these is a 'Filter for Coding Challenges:' section with a 'Categories' dropdown and a 'CLEAR' button. Underneath are several challenge cards. Each card contains the challenge name, difficulty, point, number of attempts, number of accepted solutions, and a 'SEE CHALLENGE' button.

Merge K Sorted Lists	Coing Change	Change	Reverse Integer	new challenge
Difficulty: Hard Point: 90  Number of Attempts: 11 Number of Accepted Solutions: 4  <a href="#">SEE CHALLENGE</a>	Difficulty: Medium Point: 80  Number of Attempts: 5 Number of Accepted Solutions: 3  <a href="#">SEE CHALLENGE</a>	Difficulty: Medium Point: 80  Number of Attempts: 1 Number of Accepted Solutions: 1  <a href="#">SEE CHALLENGE</a>	Difficulty: Medium Point: 75  Number of Attempts: 12 Number of Accepted Solutions: 6  <a href="#">SEE CHALLENGE</a>	Difficulty: Medium Point: 50  Number of Attempts: 3 Number of Accepted Solutions: 2  <a href="#">SEE CHALLENGE</a>
Add Two Numbers	.Median of Two Sorted Arrays	Two Sum	Sudoku Solver	Arrays
Difficulty: Medium Point: 60  Number of Attempts: 12 Number of Accepted Solutions: 3  <a href="#">SEE CHALLENGE</a>	Difficulty: Hard Point: 100  Number of Attempts: 3 Number of Accepted Solutions: 2  <a href="#">SEE CHALLENGE</a>	Difficulty: Easy Point: 40  Number of Attempts: 3 Number of Accepted Solutions: 1  <a href="#">SEE CHALLENGE</a>	Difficulty: Hard Point: 101  Number of Attempts: 7 Number of Accepted Solutions: 2  <a href="#">SEE CHALLENGE</a>	Difficulty: Hard Point: 100  Number of Attempts: 5 Number of Accepted Solutions: 2  <a href="#">SEE CHALLENGE</a>

## 2.1. Filtering for Coding Challenges

Users can filter the challenges by their categories. More than one category can be chosen for filtering. When the “Clear” button is pressed, all challenges without any filter can be reachable again.

The screenshot shows a user interface for filtering coding challenges. On the left, there's a sidebar titled "Filter for Coding Challenges" with a "Categories" dropdown menu. The dropdown is currently set to "Graph" and "BFS". Below the dropdown are several category filters: "DP" (selected), "Array", "Linked list", and "Stack". Under each filter, there are details like "Point: 80" or "Point: 101" and "Number of Attempts: 5" or "Number of Accepted Solutions: 3". To the right of the sidebar, there are four challenge cards:

- Sudoku Solver**: Difficulty: Hard, Point: 101. Details: Number of Attempts: 7, Number of Accepted Solutions: 2. [SEE CHALLENGE](#)
- Search Insert Position**: Difficulty: Easy, Point: 35. Details: Number of Attempts: 10, Number of Accepted Solutions: 4. [SEE CHALLENGE](#)
- Network Delay Time**: Difficulty: Medium, Point: 70. Details: Number of Attempts: 20, Number of Accepted Solutions: 8. [SEE CHALLENGE](#)
- presentation**: Difficulty: Hard, Point: 90. Details: Number of Attempts: 0, Number of Accepted Solutions: 0. [SEE CHALLENGE](#)

## 3. List of Non-Coding Questions

Every user can see a list of non-coding questions by clicking “Non Coding Challenges” in the menu. By pressing the “See Challenge” button, different types of details will be open for different types of users.

The screenshot shows a user interface for filtering non-coding challenges. On the left, there's a sidebar titled "Filter for Non-Coding Challenges" with a "Categories" dropdown menu. The dropdown is currently set to "Categories". Below the dropdown are several category filters: "DP : 8", "Graph : 5", "Array : 5", "Linked list : 4", "BFS : 2", and "DFS : 1". To the right of the sidebar, there are five challenge cards arranged in two rows:

Hardware	Hardware	Structers	Trees	Trees	DP
Difficulty: Easy					
<a href="#">SEE CHALLENGE</a>					
non coding	non coding	non dodinh 2	Q1 for part time	WHat is cache?	
Difficulty: Easy					
<a href="#">SEE CHALLENGE</a>					

### 3.1. Filtering for Non-Coding Questions

Users can filter the challenges by their categories. More than one category can be chosen for filtering. When the “Clear” button is pressed, all challenges without any filter can be reachable again.

The screenshot shows a user interface for filtering challenges. At the top, there's a blue header bar with the text "Welcome" and a "LOGOUT" link. Below the header, a search bar labeled "Filter for Non-Coding Challenges:" contains the text "DP". To the right of the search bar is a "CLEAR" button. A dropdown menu titled "Categories" is open, showing a list of challenge types: "Array", "Linked list", "Stack", "Graph", "DFS", and "BFS". The "DP" category is selected, indicated by a blue border around it. Below the categories, there are three columns of challenge cards. The first column contains two cards: "non coding" and "non coding". The second column contains two cards: "non dodinh 2" and "Q1 for part time". The third column contains one card: "WHat is cache?". Each card has a "Difficulty: Easy" label and a "SEE CHALLENGE" button below it.

Category	Challenge Name	Difficulty	Action
DP	non coding	Easy	SEE CHALLENGE
DP	non coding	Easy	SEE CHALLENGE
DP	non dodinh 2	Easy	SEE CHALLENGE
DP	Q1 for part time	Easy	SEE CHALLENGE
DP	WHat is cache?	Easy	SEE CHALLENGE

## 4. Details of A Contest

Every user can see details of contests. In this part different information like author, start time, end time duration etc. can be seen. Apart from them, some insight reports which are the number of participants, maximum, minimum and average points gained from the contest can be seen as well. Sponsors for the Contest are also displayed on this screen. Lastly leaderboard and results for every participant can be reachable from the table at the end of the page. From this table, by clicking the “See Profile” button, one can go to the profile page of another coder.

Weekly Contest	
<b>Author</b>	<b>Number of Participants</b>
Hakan	8
<b>Description</b>	<b>Average Point</b>
weekly Contest 16	142
<b>Difficulty</b>	<b>Max Point</b>
70	295
<b>Start Time</b>	<b>Min Point</b>
16-05-2022 12:12	0
<b>Deadline</b>	
17-05-2022 14:12	
<b>Duration</b>	<b>Sponsors</b>
180	JPMorgan Yemek Sepeti

Username	Point	Details
zehra	295	SEE PROFILE
Oğuz	245	SEE PROFILE
Sabri	235	SEE PROFILE
Ebrar	205	SEE PROFILE
Sude	160	SEE PROFILE
Sena	0	SEE PROFILE
can	0	SEE PROFILE
deniz	0	SEE PROFILE

## 5. List of Coders

List of all coders can be reachable from the menu, “Coders” part. In this page all users are displayed in order by their points that they gained from contests. Similar to the leaderboard table, one can reach the profile page of a coder by clicking to “See Profile” button.

The screenshot shows a web application interface with a blue header bar. On the left of the header is a menu icon (three horizontal lines) and the word "Welcome". On the right is a user icon and the word "LOGOUT". Below the header is a search bar with a magnifying glass icon and the placeholder text "Username". To the right of the search bar is a "CLEAR" button. The main content area contains a table with the following columns: Placement, Username, Name, Place, Position, Rating Point, and Profile. The table lists nine users:

Placement	Username	Name	Place	Position	Rating Point	Profile
1	Ebar	Ebar	Bilkent University	Student	5679	<a href="#">SEE PROFILE</a>
2	Oğuz	Oğuz	Bilkent University	Student	5296	<a href="#">SEE PROFILE</a>
3	Sabri	Sabri	JPMorgan	Intern	3144	<a href="#">SEE PROFILE</a>
4	Sude	Sude Ari	KU	Intern	2337	<a href="#">SEE PROFILE</a>
5	zehra	zehra	Bilkent University	Student	2190	<a href="#">SEE PROFILE</a>
6	Sena	Sena Genc	Bilkent University	Student	540	<a href="#">SEE PROFILE</a>
7	SelbiCoder	Selbi	Bilkent	Student	0	<a href="#">SEE PROFILE</a>
8	Mahmut	Mahmut Tuncer	Bilkent University	TA	0	<a href="#">SEE PROFILE</a>
9	AkinCoder	Akin	Yemeksepeti	Software developer	0	<a href="#">SEE PROFILE</a>

### 5.1. Search for Coders

By using the search button, all the users that include the string written in the “Username” area can be reachable. By clicking the “Clear” button all coders without any filter can be reachable again.

The screenshot shows the same web application interface as the previous one, but with a search filter applied. The search bar now contains the letter "S", and the table below shows only four users who contain "S" in their username or name:

Placement	Username	Name	Place	Position	Rating Point	Profile
1	Sabri	Sabri	JPMorgan	Intern	3144	<a href="#">SEE PROFILE</a>
2	Sude	Sude Ari	KU	Intern	2337	<a href="#">SEE PROFILE</a>
3	Sena	Sena Genc	Bilkent University	Student	540	<a href="#">SEE PROFILE</a>
4	SelbiCoder	Selbi	Bilkent	Student	0	<a href="#">SEE PROFILE</a>

# Coder

## 1. Registration

Coders can register by filling the blank areas. Any user cannot use the same username or mail that is registered with the system before. In that case, the system will show notification.

Welcome CANCEL

---

Username	Birth Year
Password	Current Position
Name	Current Company
E-mail	

REGISTER

## 2. Menu

A coder can see the Menu by clicking the menu icon at the top of the page and see the pages they can go to.

The screenshot shows a sidebar menu on the left with the following categories:

- CODING CHALLENGES
- NON CODING CHALLENGES
- NEW CONTESTS
- REGISTERED CONTESTS
- STARTED CONTESTS
- FINISHED CONTESTS
- PAST INTERVIEWS
- NEW INTERVIEWS
- CODERS
- PROFILE

The main area displays several challenge cards:

- Coins Change**: Difficulty: Medium, Point: 80. Number of Attempts: 5, Number of Accepted Solutions: 3. Buttons: SEE CHALLENGE, CLEAR.
- Change**: Difficulty: Medium, Point: 80. Number of Attempts: 1, Number of Accepted Solutions: 1. Buttons: SEE CHALLENGE, CLEAR.
- Reverse Integer**: Difficulty: Medium, Point: 75. Number of Attempts: 12, Number of Accepted Solutions: 6. Buttons: SEE CHALLENGE, CLEAR.
- Add Two Numbers**: Difficulty: Medium, Point: 60. Number of Attempts: 12, Number of Accepted Solutions: 3. Buttons: SEE CHALLENGE, CLEAR.
- .Median of Two Sorted Arrays**: Difficulty: Hard, Point: 100. Number of Attempts: 3, Number of Accepted Solutions: 2. Buttons: SEE CHALLENGE, CLEAR.

At the top right of the main area are the icons for LOGOUT and CLEAR.

### 3. Solving Coding Challenge

By clicking the “See Details” button from the list of coding questions, a coder can reach the page to solve the problem. In this page the coder can choose a programming language, write their answer in the gray area and submit their solution. When their solution is submitted a notification will be seen on the page.

The screenshot shows a web-based coding challenge interface. At the top, there's a blue header bar with the text "Welcome" and "LOGOUT". Below the header, there are three tabs: "SOLVE CHALLENGE" (which is active), "SUBMISSIONS", and "INFORMATION".  
  
The main content area has two sections:

- Description:** A section titled "Coin Change" with the following details:
  - You are given an integer array coins representing coins of different denominations and an integer amount representing a total amount of money.
  - Return the fewest number of coins that you need to make up that amount. If that amount of money cannot be made up by any combination of the coins, return -1.
  - You may assume that you have an infinite number of each kind of coin.
- Code:** A code editor with a dropdown menu "Choose Programming Language" showing Java selected. The code area contains Java code for the "Coin Change" problem, with syntax highlighting for keywords like "int", "for", and "if". The code is as follows:

```
Java
public int coinChange(int[] coins, int amount) {
    if(amount == 0)
        return 0;
    int number = -1;
    for(int i = coins.length-1; i >= 0; i--) {
        if(coins[i] <= amount){
            int newNumber =
```

At the bottom right of the code editor is a blue "SUBMIT" button.

#### 3.1 See Old Submissions

A code can see their old submission with the time they submitted and the number of failed and passed test cases by clicking the “Submission” button. In this page the coder can see details of any submission which are programming language and answer by clicking the “Details” button.

The screenshot shows a web-based submission history and details interface. At the top, there's a blue header bar with the text "Welcome" and "LOGOUT". Below the header, there are three tabs: "SOLVE CHALLENGE", "SUBMISSIONS" (which is active), and "INFORMATION".  
  
The main content area has two sections:

- Submissions:** A table showing a list of submissions with columns: Date, Time, Number of Passed Cases, Number of Failed Cases, and Details. The table contains five rows of data:

Date	Time	Number of Passed Cases	Number of Failed Cases	Details
2022-03-24	13:37	7	3	DETAILS
2022-03-24	11:45	5	5	DETAILS
2022-03-23	17:28	2	8	DETAILS
2022-03-23	16:12	0	10	DETAILS
2022-03-23	15:58	1	9	DETAILS
- Details:** A section for viewing the details of a specific submission. It includes:
  - Programming Language:** Java
  - Answer:** The code for the Java submission, which is a recursive helper function for the "Coin Change" problem. The code is as follows:

```
import java.util.Arrays;
class Solution {
    public int coinChange(int[] coins, int amount) {
        Arrays.sort(coins);
        return helper(coins, amount);
    }
    public int helper(int[] coins, int amount) {
        if(amount == 0)
            return 0;
        int number = -1;
        for(int i = coins.length-1; i >= 0; i--) {
            if(coins[i] <= amount){
                int newNumber = coinChange(coins, amount-coins[i]); // 6, 1
                if(newNumber != -1)
                    if(newNumber < number || number == -1)
                        return number = newNumber + 1;
            }
        }
    }
}
```

### 3.2 See Details of Coding Challenge

A coder can see details of a challenge like author, number of total attempts and the number of accepted ones ect. by clicking the “Information” button.

Welcome

SOLVE CHALLENGE SUBMISSIONS INFORMATION

**Title**  
Coin Change

**Author**  
Hakan

**Categories**

- Graph
- Linked list

**Difficulty**  
Medium

**Point**  
80

**Number of Attempts**  
5

**Number of Accepted Answers**  
3

LOGOUT

### 4. Solving Non-Coding Question

By clicking the “See Details” button from the list of non coding questions, a coder can reach the page to give an answer to the problem. In this page the coder can write their answer in the gray area and submit their answer. When their solution is submitted a notification will be seen on the page.

Welcome

SOLVE CHALLENGE SUBMISSION INFORMATION OTHER ANSWERS

**Description:**

**Structers**

Which strucuter should be used if there are lost of add and remove

**Answer:**

SUBMIT

LOGOUT

#### 4.1. See Your Submission

For non coding challenges, only one submission is allowed. After their first submission, when they come to this page again, they will see their answer in this page as well and not be able to write or change anything in the gray area.

The screenshot shows a web interface for a challenge titled "Structers". The top navigation bar includes "Welcome", "LOGOUT", and tabs for "SOLVE CHALLENGE", "SUBMISSION", "INFORMATION", and "OTHER ANSWERS". The "SUBMISSION" tab is currently active, indicated by a red underline. The main content area has two sections: "Description:" and "Answer:". The "Description:" section contains the question: "Which structer should be used if there are lost of add and remove". The "Answer:" section displays a grayed-out area with the text "Submission Date:17-05-2022 11:16" and "Linked List". A "SUBMIT" button is located at the bottom right of the answer area.

#### 4.2 See Details of Non Coding Question

Similar to the coding challenge information, a coder can see detailed information of a non coding challenge by clicking “Information” button.

The screenshot shows a web interface for a challenge titled "Structers". The top navigation bar includes "Welcome", "LOGOUT", and tabs for "SOLVE CHALLENGE", "SUBMISSION", "INFORMATION", and "OTHER ANSWERS". The "INFORMATION" tab is currently active, indicated by a red underline. The main content area displays several details about the challenge:

- Title:** Structers
- Author:** Hakan
- Categories:** DP, Linked list
- Difficulty:** Easy

#### 4.3 See Others' Answer

Once a coder submits an answer to a non coding challenge, they become able to see others' answers. By clicking the "Other Answers" button a coder can see other answers. If the coder has not answered the question a warning will be displayed and other answers will not be seen on this screen.

The screenshot shows a user interface for viewing other answers. At the top, there's a blue header bar with the word 'Welcome' and a 'LOGOUT' button. Below the header, there are tabs: 'SOLVE CHALLENGE', 'SUBMISSION', 'INFORMATION', and 'OTHER ANSWERS', with 'OTHER ANSWERS' being the active tab. The main content area displays four user profiles in separate boxes:

- zehra**: Profile picture of a person, linked list.
- Ebrar**: Profile picture of a person, Heap.
- Sena**: Profile picture of a person, Linked List.
- Sude**: Profile picture of a person.

## 5. Contests

### 5.1 Future and Non Registered Contests

A coder can see the list of future contests that they have not registered by clicking "New Contests" button from the menu.

The screenshot shows a table of contests. At the top, there's a blue header bar with the word 'Welcome' and a 'LOGOUT' button. Below the header, there's a table with columns: Title, Difficulty, Start Time, Deadline, Duration, and Details. The table contains the following data:

Title	Difficulty	Start Time	Deadline	Duration	Details
IEEE Contest	70	19-05-2022 10:21	20-05-2022 10:21	120	DETAILS
presnetation	50	19-05-2022 11:10	20-05-2022 11:10	100	DETAILS
Conteste Hosgeldiniz	10	19-05-2022 06:14	20-05-2022 03:14	120	DETAILS
title	5	23-05-2022 18:25	23-08-2022 18:25	60	DETAILS

#### 5.1.1. Register to A Contest

When the “Details” button of a contest is clicked, the description of the contest will be displayed. By clicking the “Register” button here, a coder can register for the contest. After registration, the coder can see the contest in “Registered Contests” part in the menu.

The screenshot shows a modal window titled "Contest Detail" with the following content:

- IEEE Contest**
- ieee is arranging a new contest
- [CLOSE](#)   [REGISTER](#)

The modal is centered over a table of contests. One row in the table is highlighted with a light gray background, corresponding to the contest shown in the modal. The table columns are: Title, Difficulty, Start Time, Deadline, Duration, and Details.

Title	Difficulty	Start Time	Deadline	Duration	Details
IEEE Contest	70	19-05-2022 10:21	20-05-2022 10:21	120	<a href="#">DETAILS</a>
presnetation	50	19-05-2022 11:10	20-05-2022 11:10	100	<a href="#">DETAILS</a>
Conteste Hosgeldiniz	10	19-05-2022 06:14	20-05-2022 03:14	120	<a href="#">DETAILS</a>
title	5	23-08-2022 18:25		60	<a href="#">DETAILS</a>

## 5.2 List of Future and Registered Contests

A coder can see the list of future contests that they have registered by clicking “Registered Contests” button from the menu.

The screenshot shows a table of registered contests. The table columns are: Title, Difficulty, Start Time, Deadline, Duration, and Details.

Title	Difficulty	Start Time	Deadline	Duration	Details
title	3	19-05-2022 06:14	20-05-2022 03:14	120	<a href="#">DETAILS</a>
Conteste Hosgeldiniz	10	19-05-2022 06:14	20-05-2022 03:14	120	<a href="#">DETAILS</a>

### 5.2.1. Cancel Registration

When the “Details” button of a contest is clicked, the description of the contest will be displayed. By clicking the “Cancel Registration” button here, a coder can cancel their registration for the contest.

The screenshot shows a user interface for managing contests. At the top, there's a dark blue header with the word "Welcome". On the right side of the header are a bell icon and a "LOGOUT" link. Below the header is a table with columns: Title, Difficulty, Start Time, Deadline, Duration, and Details. There are two rows of data in the table. The first row has a "Title" column value of "title", a "Difficulty" column value of "3", a "Start Time" column value of "19-05-2022 06:14", a "Deadline" column value of "20-05-2022 03:14", a "Duration" column value of "120", and a "Details" column value of "DETAILS". The second row has a "Title" column value of "Conteste Hosgeldiniz", a "Difficulty" column value of "10", a "Start Time" column value of "19-05-2022 06:14", a "Deadline" column value of "20-05-2022 03:14", a "Duration" column value of "120", and a "Details" column value of "DETAILS". A modal dialog box is overlaid on the page, centered over the second contest row. The modal has a white background and a thin black border. It contains the text "Contest Detail" at the top, followed by the bolded title "Conteste Hosgeldiniz", a subtitle "Hakanin 2 Contest", and two buttons at the bottom: "CLOSE" and "CANCEL REGISTRATION".

Title	Difficulty	Start Time	Deadline	Duration	Details
title	3	19-05-2022 06:14	20-05-2022 03:14	120	<a href="#">DETAILS</a>
Conteste Hosgeldiniz	10	19-05-2022 06:14	20-05-2022 03:14	120	<a href="#">DETAILS</a>

### 5.3 List of Started Contests

List of contests that has started and the coder has registered can be seen by clicking the “Started Contests” button in the menu.

The screenshot shows a list of contests that have started. At the top, there's a dark blue header with the word "Welcome". On the right side of the header are a bell icon and a "LOGOUT" link. Below the header is a table with columns: Title, Difficulty, Start Time, Deadline, Duration, and See Details and Start. There are three rows of data in the table. The first row has a "Title" column value of "Weekly Contest", a "Difficulty" column value of "70", a "Start Time" column value of "16-05-2022 12:12", a "Deadline" column value of "17-05-2022 14:12", a "Duration" column value of "180", and a "See Details and Start" column value of "DETAILS". The second row has a "Title" column value of "ACM Contest", a "Difficulty" column value of "75", a "Start Time" column value of "16-05-2022 12:00", a "Deadline" column value of "23-05-2022 12:00", a "Duration" column value of "240", and a "See Details and Start" column value of "DETAILS". The third row has a "Title" column value of "IEEE", a "Difficulty" column value of "80", a "Start Time" column value of "17-05-2022 14:09", a "Deadline" column value of "24-05-2022 14:09", a "Duration" column value of "120", and a "See Details and Start" column value of "DETAILS".

Title	Difficulty	Start Time	Deadline	Duration	See Details and Start
Weekly Contest	70	16-05-2022 12:12	17-05-2022 14:12	180	<a href="#">DETAILS</a>
ACM Contest	75	16-05-2022 12:00	23-05-2022 12:00	240	<a href="#">DETAILS</a>
IEEE	80	17-05-2022 14:09	24-05-2022 14:09	120	<a href="#">DETAILS</a>

#### 5.3.1. Start to Solve A contest

When the “Details” button of a contest is clicked, the description of the contest will be displayed. By clicking the “See Leaderboard” button here, a coder can go to the details of the contest and see the leaderboard and other details there. If the coder presses the “start” button, their start time will be saved and they will go to list of questions for the contest.

The screenshot shows a web-based contest management system. At the top, there's a navigation bar with a menu icon, "Welcome", a notification bell icon, and "LOGOUT". Below this is a table listing three contests:

Title	Difficulty	Start Time	Deadline	Duration	Action
Weekly Contest	70	16-05-2022 12:12	17-05-2022 14:12	180	<a href="#">DETAILS</a>
ACM Contest	75	16-05-2022 12:00	23-05-2022 12:00	240	<a href="#">DETAILS</a>
IEEE	80	17-05-2022 14:09	24-05-2022 14:09	120	<a href="#">DETAILS</a>

A modal window titled "Contest Detail" is open for the "IEEE" contest. It contains the title "IEEE", a brief description "This year's IEEE contest", and three buttons: "CANCEL", "SEE LEADERBOARD", and "START".

### 5.3.2. Solve Contest

After they press the “Start” button they can see the list of questions and solve them until duration time is up. After duration time is up they will not be able to make new submissions. Solving question page is same as the Solving Coding Challenge page.

The screenshot shows a list of five coding challenges under a "Contest" section:

- Network Delay Time**: Difficulty: Medium, Point: 70. Number of Attempts: 20, Number of Accepted Solutions: 8. Buttons: SEE CHALLENGE, FINISH.
- Two Sum**: Difficulty: Easy, Point: 40. Number of Attempts: 3, Number of Accepted Solutions: 1. Buttons: SEE CHALLENGE.
- Add Two Numbers**: Difficulty: Medium, Point: 60. Number of Attempts: 12, Number of Accepted Solutions: 3. Buttons: SEE CHALLENGE.
- Reverse Integer**: Difficulty: Medium, Point: 75. Number of Attempts: 12, Number of Accepted Solutions: 6. Buttons: SEE CHALLENGE.
- Merge k Sorted Lists**: Difficulty: Hard, Point: 90. Number of Attempts: 11, Number of Accepted Solutions: 4. Buttons: SEE CHALLENGE.

### 5.3.3. After Your Duration Time is Up

Before the duration is up, if the coder comes back to “Started Contests” they will see a “Continue” button and be able to solve questions. After the duration time is up, a coder cannot see the “Start” or “Continue” button anymore.

The screenshot shows a user interface for managing contests. At the top, there's a dark blue header bar with the word "Welcome" and a menu icon on the left, and "LOGOUT" on the right. Below this is a table with columns: Title, Difficulty, Start Time, Deadline, Duration, and a "See Details and Start" button. Three rows are listed: "Weekly Contest" (Difficulty 70), "ACM Contest" (Difficulty 75), and "IEEE" (Difficulty 80). The "Weekly Contest" row has a "See Details and Start" button which is currently active, displaying a modal window. The modal has a title "Contest Detail" and a sub-section "Weekly Contest" with the text "weekly Contest 16". It also contains "CANCEL" and "SEE LEADERBOARD" buttons. The background of the main table is dimmed.

Title	Difficulty	Start Time	Deadline	Duration	
Weekly Contest	70	16-05-2022 12:12	17-05-2022 14:12	180	<a href="#">DETAILS</a>
ACM Contest	75	16-05-2022 12:00	23-05-2022 12:00	240	<a href="#">DETAILS</a>
IEEE	80	17-05-2022 14:09	24-05-2022 14:09	120	<a href="#">DETAILS</a>

#### 5.4. List of Finished Contests

A coder can see a list of contests that their deadline has passed by clicking the “Finished Contests” button from the menu. They can go to the details of a contest from here.

The screenshot shows a user interface for viewing finished contests. At the top, there's a dark blue header bar with the word "Welcome" and a menu icon on the left, and "LOGOUT" on the right. Below this is a table with columns: Title, Difficulty, Start Time, Deadline, Duration, and a "Details" button. One row is listed: "Weekly Contest" (Difficulty 70). The "Details" button for this row is active, displaying a modal window. The modal has a title "Contest Detail" and a sub-section "Weekly Contest" with the text "weekly Contest 16". It also contains "CANCEL" and "SEE LEADERBOARD" buttons. The background of the main table is dimmed.

Title	Difficulty	Start Time	Deadline	Duration	
Weekly Contest	70	16-05-2022 12:12	17-05-2022 14:12	180	<a href="#">SEE DETAILS</a>

## 6. See Other Coders’s Profile

A coder can see the profile page of another coder by clicking the “See Profile” button from “Coders” part in the menu or leaderboard tables in Contest Details. In this page they can see

how many contests a coder attends and what is the result of these contests. Moreover, they can see who gives a referral to that coder.

The screenshot shows a user profile for 'Ebrar' (Student). On the left, there's a large circular placeholder for a profile picture. Below it, the name 'Ebrar' and 'Student' are displayed, along with a 'GIVE REFERRAL' button. To the right, three boxes show statistics: '5679 Ranking', '2 Contest', and '2 Questions'. Below these, sections for 'Awards' and 'Referrals' are shown. The 'Awards' section lists 'Weekly Contest' (4) and 'ACM Contest' (1). The 'Referrals' section shows a referral to 'SelbiCoder' (Student) with the note 'She is a good coder'.

### 6.1. Give Referral to Another Coder

A coder can give a referral to another coder. They can give only one referral for a coder and they have to write an explanation to indicate why they give the referral.

The screenshot shows the same user profile for 'Ebrar' (Student). A modal window titled 'Give referral' is open in the center. It contains a text input field with the placeholder 'Please write your reson to give referral' and a single-line text entry 'She was my teammate and is a very responsible person'. At the bottom of the modal are 'CANCEL' and 'SUBMIT' buttons. The background of the page is dimmed to indicate an active modal.

### 7. See Your own Profile

A coder can see their own profile by clicking the "Profile" button from the menu. In this part they can see their information as well.

The screenshot shows a user profile page with a blue header bar. On the left, there is a large circular placeholder for a profile picture with the text "Sabri Intern" below it. To the right, three boxes show statistics: "3144 Ranking", "4 Contest", and "4 Questions". Below these are sections for "Awards" and "Referrals". The "Awards" section lists "Weekly Contest" (3) and "title" (1). The "Referrals" section shows a profile for "SelbiCoder Student" with the note "He/She is good".

## 8. See Notifications

By clicking the notification icon at the top of the page, a coder can see their notifications. When an interview results in, the system sends a notification to the user.

The screenshot shows a notifications page with a blue header bar. It displays two notifications about interviews: one for "Weak Hire" on 2022-05-16T16:19:08Z and another for "Strong No Hire" on 2022-05-17T08:20:10Z. Both notifications include the message "You found [status] in one of the interviews check your interviews".

## 9. See Interviews

Coders can view their list of ongoing, past and future interviews.

## 9.1 See Old Interviews

By clicking ‘Past Interviews’ in the menu, coder users can view the past interviews they attended. They will see a table of interviews. They can see the name of the company, duration of the interview, start time, end time, and result of the interview. If the interview result is not determined by the company yet, then ‘Not determined’ will be displayed.

The screenshot shows a web application interface. At the top, there is a blue header bar with the text "Welcome" on the left and "LOGOUT" on the right. Below the header, the title "Past Interviews" is centered. A table follows, displaying three rows of interview data. The columns are labeled: Company, Duration, Position, Start time, End time, and Result. The data is as follows:

Company	Duration	Position	Start time	End time	Result
company1	0	Part time	16-05-2022 19:29	17-05-2022 03:16	Strong Hire
company1	0	Part time fall	16-05-2022 03:23	17-05-2022 03:03	Strong No Hire
company1	60	Summer internship	17-05-2022 03:11	25-05-2022 03:11	Not determined

## 9.2 See New Interviews

Coders can view the list of future and ongoing interviews they are attending. They can see the company name, the position they are applying for, the interview duration, the interview's starting time, and the ending time of the interview. They can choose ‘Start’ and start the

interview from this list. For interviews that have not started yet, ‘Not started yet’ will be displayed.

The screenshot shows a 'New Interviews' page with a search interface at the top. It includes fields for 'Start Date' (dd.mm.yyyy) and 'End Date' (dd.mm.yyyy), both with calendar icons. Below these are 'Start Time' and 'End Time' fields, each with a time picker icon. To the right of the time fields are 'SEARCH' and 'CLEAR' buttons. A table below the search fields lists three interview entries:

Company	Duration	Position	Start time	End time	Link
company1	0	Summer intern	23-05-2022 18:25	23-08-2022 18:25	Not started yet
company1	56	Summer intern	25-05-2022 03:13	31-05-2022 03:10	Not started yet
company1	60	Summer internship	17-05-2022 03:11	25-05-2022 03:11	<button>START</button>

### 9.3 Filter New Interviews

Coders can select a date and time range and choose to search for new interviews within that range.

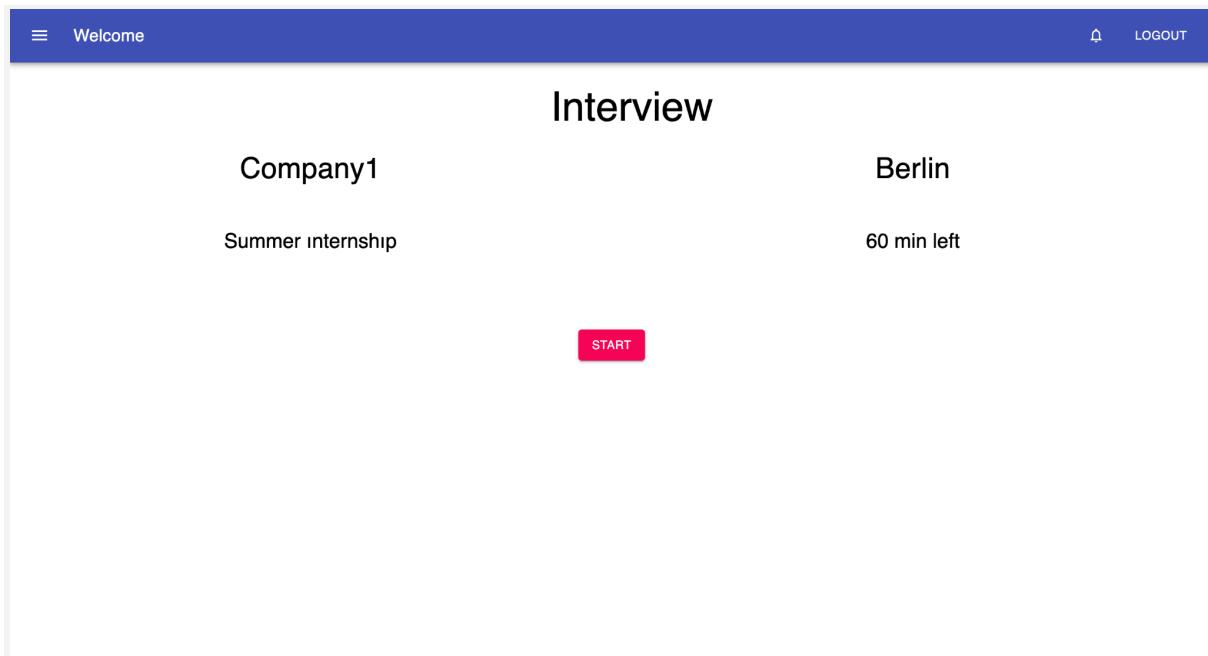
The screenshot shows a 'New Interviews' page with a search interface at the top. It includes fields for 'Start Date' (10.05.2022) and 'End Date' (22.05.2022), both with calendar icons. Below these are 'Start Time' and 'End Time' fields, each with a time picker icon. To the right of the time fields are 'SEARCH' and 'CLEAR' buttons. A table below the search fields lists one interview entry:

Company	Duration	Position	Start time	End time	Link
company1	60	Summer internship	17-05-2022 03:11	25-05-2022 03:11	<button>START</button>

## 10. Attend Interview

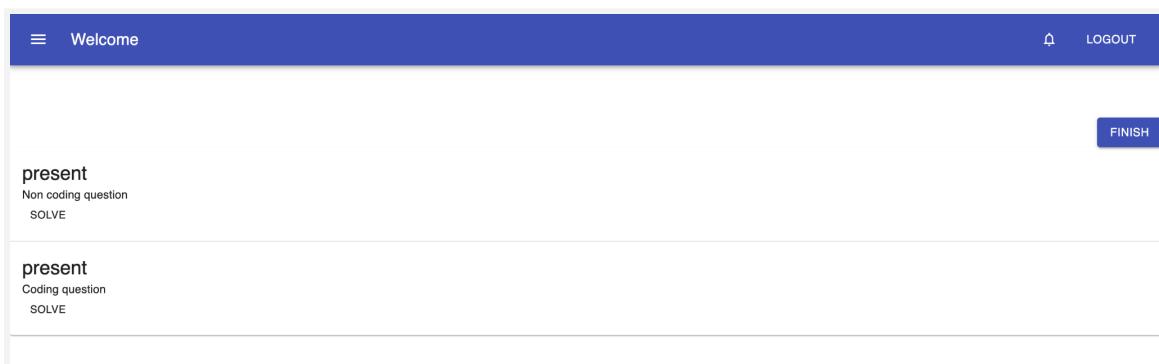
### 10.1 Start Interview

After the coder chooses to start an interview, they will be taken to the start interview page, where they can view the company name, the company location, and the duration of the interview. By clicking the “start”, they start attending the interview.



### 10.2 Attend Interview

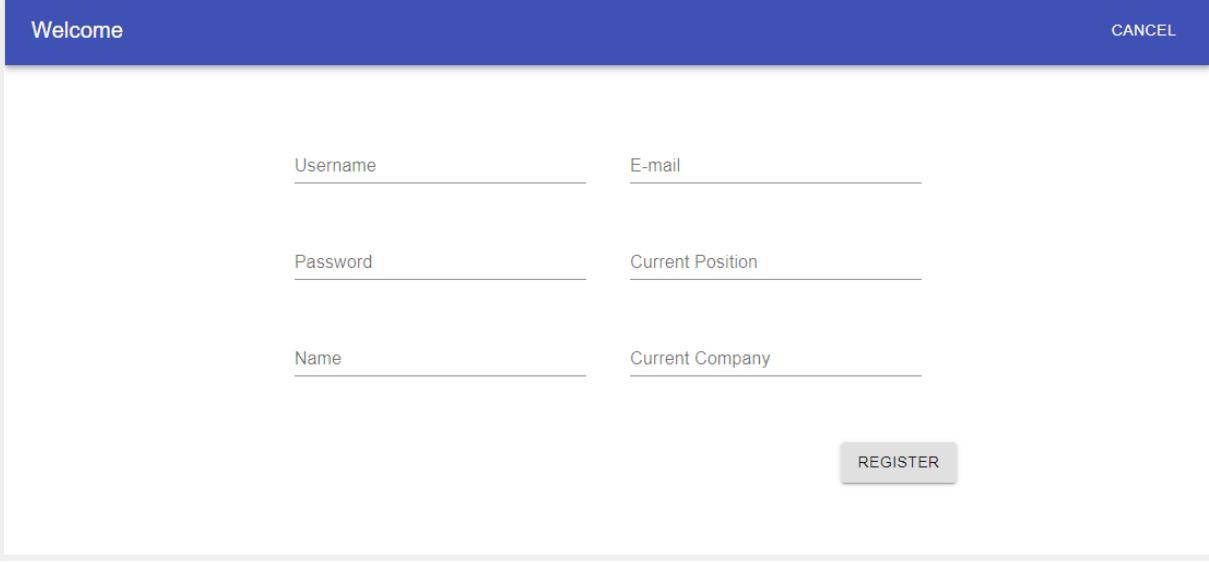
When coders attend the interview, they will see a list of questions. They can choose to solve them. In that case, they will be redirected to a solving page, and they can solve or answer the question just the way they would for non-interview questions. After the coder has finished solving and answering the questions, they can choose to finish the interview.



# Editor

## 1. Register

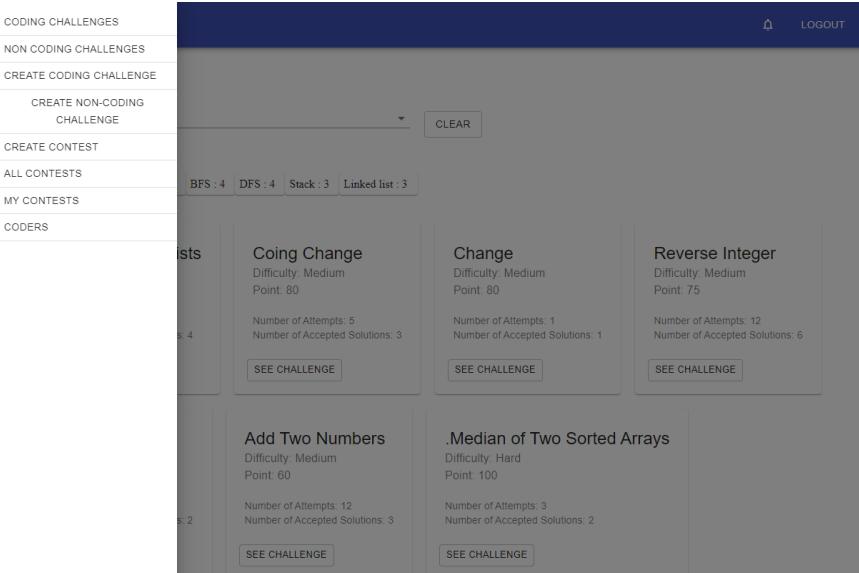
Editors can register by filling the blank areas. Any user cannot use the same username or mail that is registered with the system before. In that case, the system will show notification.



The registration form is titled "Welcome" at the top left and has a "CANCEL" button at the top right. It contains four input fields arranged in a 2x2 grid: "Username" and "E-mail" in the top row, and "Password" and "Current Position" in the bottom row. Below these is another 2x2 grid: "Name" and "Current Company". At the bottom right is a "REGISTER" button.

## 2. Menu

An editor can see the Menu by clicking the menu icon at the top of the page and see the pages they can go to.



The menu interface shows a sidebar with navigation links: CODING CHALLENGES, NON CODING CHALLENGES, CREATE CODING CHALLENGE, CREATE NON-CODING CHALLENGE, CREATE CONTEST, ALL CONTESTS, MY CONTESTS, and CODERS. The main area displays a grid of challenges and contests. The challenges shown are:

- Coing Change (Difficulty: Medium, Point: 80)
- Change (Difficulty: Medium, Point: 80)
- Reverse Integer (Difficulty: Medium, Point: 75)
- Add Two Numbers (Difficulty: Medium, Point: 60)
- .Median of Two Sorted Arrays (Difficulty: Hard, Point: 100)

Below each challenge is a "SEE CHALLENGE" button. At the bottom of the sidebar, it says "Number of Contests: 4" and "Number of Challenges: 5".

### 3. See Coding Challenge

By clicking the “See Details” button from the list of coding questions, an editor can reach the page that displays information of the challenge like author, question, categories, solution, test cases etc. Apart from them number of total attempts and the number of accepted ones can be seen as well.

The screenshot shows a web-based application interface for viewing a coding challenge. At the top, there's a blue header bar with the text "Welcome" on the left and "LOGOUT" on the right. Below the header, the page is divided into two main sections: "Description" on the left and "Author" on the right. The "Description" section contains the problem statement for the "Change" challenge, which involves finding the fewest number of coins to make up a given amount. It includes sample inputs (1,2,3,4,5) and outputs (3). The "Author" section shows the challenge was created by "Hakan". To the right of the "Description" section, there are sections for "Categories" (Graph), "Difficulty" (Medium), "Point" (80), "Number of Attempts" (1), and "Number of Accepted Answers" (1). On the far left, there are sections for "Solution" (main) and "Test Case Inputs" (1,2,3,4,5).

### 4. See Non Coding Question

By clicking the “See Details” button from the list of non coding questions, an editor can reach the page that displays information of the non coding challenge like author, question, categories etc..

The screenshot shows a web-based application interface for viewing a non-coding challenge. At the top, there's a blue header bar with the text "Welcome" on the left and "LOGOUT" on the right. Below the header, the page is divided into two main sections: "Description" on the left and "Author" on the right. The "Description" section contains the problem statement for the "Structers" challenge, which asks for the appropriate structure to use if there are lots of add and remove operations. It includes sample inputs (Which strucer should be used if there are lots of add and remove) and outputs (Easy). The "Author" section shows the challenge was created by "Hakan". To the right of the "Description" section, there are sections for "Categories" (DP, Linked list), "Difficulty" (Easy), and "Point" (80). On the far left, there are sections for "Solution" (main) and "Test Case Inputs" (Which strucer should be used if there are lots of add and remove).

## 5. Create Coding Challenge

An editor can create a coding challenge. They can reach this page by clicking the “Create Coding Challenge” button from the menu.

The screenshot shows a web application interface for creating a coding challenge. At the top, there's a blue header bar with the text "Welcome" and "LOGOUT". Below the header, the main title is "Create new coding challenge". The form fields include:

- Title:** Challenge 100
- Question:** demo question
- Points:** 50
- Difficulty:** A dropdown menu showing "Easy" (selected), "Medium", and "Hard".
- Solution:** An empty text area.
- Categories:** A dropdown menu showing "Input testcase" and "Output testcase".
- Buttons:** "CANCEL" and "SAVE" at the bottom.

## 6. Create Non-Coding Question

An editor can create a non coding question. They can reach this page by clicking the “Create Non-Coding Challenge” button from the menu.

The screenshot shows a web application interface for creating a non-coding question. At the top, there's a blue header bar with the text "Welcome" and "LOGOUT". Below the header, the main title is "Create new question". The form fields include:

- Title:** Non Coding Question 100
- Question:** demo question
- Points:** 20
- Difficulty:** A dropdown menu showing "Easy" (selected).
- Categories:** A dropdown menu showing "Linked list", "Array", and "Category".
- Buttons:** "CANCEL" and "SAVE" at the bottom.

## 7. Create Contest

An editor can create a contest. They can reach this page by clicking the “Create Contest” button from the menu. When they are creating a new contest, they cannot create a contest that has earlier deadline than start time.

Welcome

Logout

### Create New Contest

Title: Newest Contest

Description: The new Contest

Difficulty: 70

Duration: 2 hour(s)

Start Date: 24.05.2022 16:30

Deadline: 31.05.2022 15:30

CREATE

15	30
16	31
17	32
18	33
19	34
20	35
21	36

### 7.1. Add Challenges to Contest

After the contest is created, the editor can choose questions that they have created to add to the contest.

ID	Title	Difficulty	Point
3	Network Delay ...	Medium	70
4	Two Sum	Easy	40
5	Add Two Num...	Medium	60
7	Reverse Integer	Medium	75
8	Merge k Sorte...	Hard	90

2 rows selected

1–5 of 27 < >

SAVE CANCEL

## 8. Contests

### 8.1 List of All Contests

Editors can see all contests by clicking the “All Contests” button from the menu. They can see details of a contest by clicking the “Details” button.

TITLE	EDITOR	IS FINISHED	DETAILS
title	Hakan Yilmaz	Not Finished	<a href="#">SEE DETAILS</a>
Newes Contest	Meryem	Not Finished	<a href="#">SEE DETAILS</a>
IEEE	Hakan Yilmaz	Not Finished	<a href="#">SEE DETAILS</a>
ACM Contest	Hakan Yilmaz	Not Finished	<a href="#">SEE DETAILS</a>
presnetation	Hakan Yilmaz	Not Finished	<a href="#">SEE DETAILS</a>
IEEE Contest	Hakan Yilmaz	Not Finished	<a href="#">SEE DETAILS</a>
Weekly Contest	Hakan Yilmaz	Not Finished	<a href="#">SEE DETAILS</a>
Contest1	Hakan Yilmaz	Finished	<a href="#">SEE DETAILS</a>
Summer Contest	Meryem	Finished	<a href="#">SEE DETAILS</a>

### 8.2. List of Your Contests

Editors can see the contests that they have created by clicking the “My Contests” button from the menu. They can see details of a contest by clicking the “Details” button.

Title	Editor	Is Finished	Details
Newes Contest	Meryem	Not Finished	<a href="#">SEE DETAILS</a>
Summer Contest	Meryem	Finished	<a href="#">SEE DETAILS</a>

## 9. See Coder's Profile

An editor can see the profile page of a coder by clicking the “See Profile” button from “Coders” part in the menu or leaderboard tables in Contest Details. In this page they can see how many contests a coder attends and what is the result of these contests. Moreover, they can see who gives a referral to that coder.

The screenshot shows the profile page for a coder named Ebrar. At the top, there is a large circular placeholder for a profile picture. Below it, the name "Ebrar" and the title "Student" are displayed. A blue button labeled "GIVE REFERRAL" is visible. To the right, three boxes show statistics: "5679 Ranking", "2 Contest", and "2 Questions". Below these, sections for "Awards" and "Referrals" are shown. The "Awards" section lists "Weekly Contest" (4) and "ACM Contest" (1). The "Referrals" section shows a referral from "SelbiCoder" with the note "She is a good coder".

### 9.1 Give Referral to A Coder

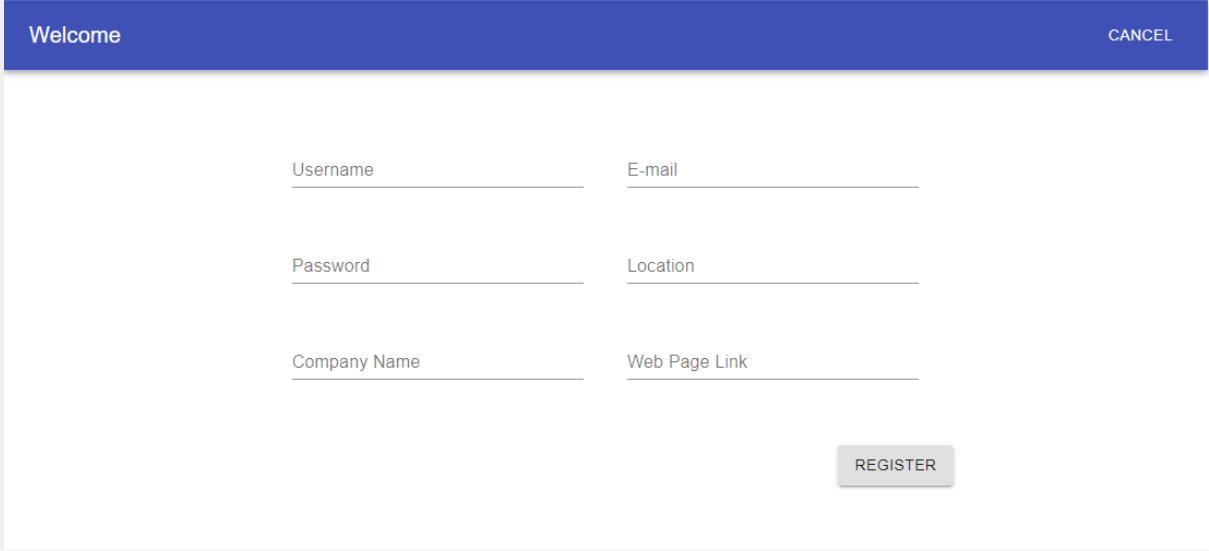
An editor can also give a referral to a coder. They can give only one referral for a coder and they have to write an explanation to indicate why they give the referral.

The screenshot shows the "Give referral" modal dialog. It has a text input field asking "Please write your reson to give referral" and two buttons at the bottom: "CANCEL" and "SUBMIT". The background of the page is dimmed, and the "Referrals" section is partially visible, showing the same information as the previous screenshot.

# Company

## 1. Register

Companies can register by filling the blank areas. Any user cannot use the same username or mail that is registered with the system before. In that case, the system will show notification.



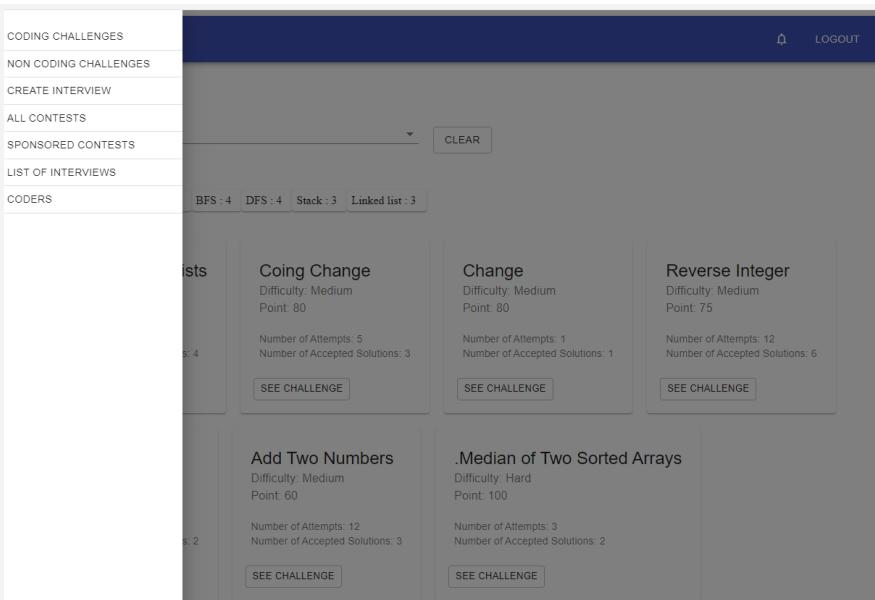
The registration form consists of four input fields arranged in a 2x2 grid. The top row contains 'Username' and 'E-mail'. The bottom row contains 'Password' and 'Location'. Below the grid is a 'Company Name' field and a 'Web Page Link' field. At the bottom right is a 'REGISTER' button.

Username	E-mail
Password	Location
Company Name	Web Page Link

REGISTER

## 2. Menu

A company can see the Menu by clicking the menu icon at the top of the page and see the pages they can go to.



The menu interface shows a sidebar with navigation links: CODING CHALLENGES, NON CODING CHALLENGES, CREATE INTERVIEW, ALL CONTESTS, SPONSORED CONTESTS, LIST OF INTERVIEWS, and CODERS. The main area displays a grid of challenges. Each challenge card includes a title, difficulty level, point value, attempt count, accepted solution count, and a 'SEE CHALLENGE' button.

Challenge Type	Title	Difficulty	Point	Attempts	Accepted Solutions
LISTS	Coing Change	Medium	80	5	3
	Change	Medium	80	1	1
	Reverse Integer	Medium	75	12	6
COPYS	Add Two Numbers	Medium	60	12	3
	Median of Two Sorted Arrays	Hard	100	3	2
	Reverse String	Medium	70	10	5

### 3. See Coding Challenge

By clicking the “See Details” button from the list of coding questions, a company can reach the page that displays information of the challenge like author, question, categories, solution, test cases ect. Apart from them number of total attempts and the number of accepted ones can be seen as well.

The screenshot shows a web application interface for a coding challenge. At the top, there is a blue header bar with the text "Welcome" on the left and "LOGOUT" on the right. Below the header, the page is divided into two main sections: "Description" and "Solution".

**Description:**

**Change**

You are given an integer array coins representing coins of different denominations and an integer amount representing a total amount of money.

Return the fewest number of coins that you need to make up that amount. If that amount of money cannot be made up by any combination of the coins, return -1.

You may assume that you have an infinite number of each kind of coin.

**Solution:**

```
main
```

**Test Case Inputs:**

[1,2,3,4,5]

**Test Case Outputs:**

3

**Author**

Hakan

**Categories**

- Graph

**Difficulty**

Medium

**Point**

80

**Number of Attempts**

1

**Number of Accepted Answers**

1

### 4. See Non Coding Question

By clicking the “See Details” button from the list of non coding questions, a company can reach the page that displays information of the non coding challenge like author, question, categories ect..

The screenshot shows a web application interface for a non-coding question. At the top, there is a blue header bar with the text "Welcome" on the left and "LOGOUT" on the right. Below the header, the page is divided into two main sections: "Description" and "Solution".

**Description:**

**Structures**

Which structure should be used if there are lots of add and remove

**Author**

Hakan

**Categories**

- DP
- Linked list

**Difficulty**

Easy

## 5. Contests

### 5.1. List of All Contests

Companies can see all contests by clicking the “All Contests” button from the menu. They can see details of a contest by clicking the “Details” button.

Welcome		Logout	
Title	Editor	Is Finished	Details
title	Hakan Yilmaz	Not Finished	SEE DETAILS
Newes Contest	Meryem	Not Finished	SEE DETAILS
IEEE	Hakan Yilmaz	Not Finished	SEE DETAILS
ACM Contest	Hakan Yilmaz	Not Finished	SEE DETAILS
presnetation	Hakan Yilmaz	Not Finished	SEE DETAILS
IEEEE Contest	Hakan Yilmaz	Not Finished	SEE DETAILS
Weekly Contest	Hakan Yilmaz	Not Finished	SEE DETAILS
Contest1	Hakan Yilmaz	Finished	SEE DETAILS
Summer Contest	Meryem	Finished	SEE DETAILS

### 5.2. Sponsored Contests

Companies can see the contests that they have sponsored by clicking the “Sponsored Contests” button from the menu. They can see details of a contest by clicking the “Details” button.

Welcome		Logout	
Title	Editor	Is Finished	Details
title	Hakan Yilmaz	Not Finished	SEE DETAILS
ACM Contest	Hakan Yilmaz	Not Finished	SEE DETAILS
Weekly Contest	Hakan Yilmaz	Not Finished	SEE DETAILS

## 6. Be Sponsor to A Contest

When a company displays the details of a contest, the “Be “Sponsor” button will be shown as well. Companies can sponsor the contest by clicking this button. If they have already sponsored a warning notification will be displayed.

The screenshot shows a contest titled "ACM Contest". It includes the following details:

Author	Number of Participants
Hakan	8

**Description:** ACM organized

**Difficulty:** 75

**Start Time:** 16-05-2022 12:00

**Deadline:** 23-05-2022 12:00

**Duration:** 240

**Sponsors:** Yemek Sepeti

**Average Point:** 95

**Max Point:** 296

**Min Point:** 0

**BE SPONSOR** button

## 7. See Profile of A Coder

A company can see the profile page of another coder by clicking the “See Profile” button from “Coders” part in the menu or leaderboard tables in Contest Details. In this page they can see how many contests a coder attends and what is the result of these contests. Moreover, they can see who gives a referral to that coder. Companies cannot give referral for coders.”

The screenshot shows the profile of a coder named "Sabri" (Intern). It includes the following information:

- Profile Picture:** Placeholder icon.
- Username:** Sabri
- Title:** Intern
- Ranking:** 3144
- Contest:** 4
- Questions:** 4
- Awards:** Weekly Contest (3)
- Referrals:** SelbiCoder (Student) - He/She is good

## 8. Create Interview

### 8.1 Create Interview

Companies can create interviews by specifying the position and duration of the interview.

The screenshot shows a web-based application for creating a new interview. At the top, there's a blue header bar with the text "Welcome" on the left and "LOGOUT" on the right. Below the header, the main title "Create new interview" is centered. The form consists of several input fields: a "Position" field containing "Position", a "Duration" field showing "0 min", and a "CREATE" button below them. At the bottom of the form, there are two buttons: "ADD CODING QUESTION" and "ADD NON CODING QUESTION". Below the form, there are two input fields labeled "Type" and "Title". At the very bottom, there are "CANCEL" and "SUBMIT" buttons.

### 8.2 Add Coding Questions to Interview

By clicking “Add Coding Question” they can create a new coding question to add to the interview.

This screenshot shows the same "Create new interview" interface as the previous one, but with different input values. The "Position" field now contains "Demo", and the "Duration" field shows "45 min". The rest of the interface, including the "CREATE" button, "ADD CODING QUESTION" and "ADD NON CODING QUESTION" buttons, the "Type" and "Title" fields, and the "CANCEL" and "SUBMIT" buttons at the bottom, remain the same.

## Create new coding challenge

Form fields for creating a new coding challenge:

- Title
- Question
- Points: Easy ▾
- Solution
- Catogories
- Duration: min ▾
- Input testcase
- Output testcase

### 8.3 Add Non-Coding Questions to Interview

By clicking “Add Non Coding Question” they can create a new non coding question to add to the interview.

Welcome Logout

### Create new interview

Position: Demo

Duration: 45 min ▾

**CREATE**

**ADD CODING QUESTION** **ADD NON CODING QUESTION**

Type Title

**CANCEL** **SUBMIT**

## Create new question

Title

Question

Points

Catogories

### 8.4. Submit Interview

After adding coding and non-coding questions to the interview, a company can click submit to move on to adding participants to the interview.

Welcome

## Create new interview

Position

Duration  min

Type	Title
Non Coding Question	Demo non coding
Coding question	Demo coding

## 8.4 Add Participants to Interview

Companies can add coders to interview as candidates by specifying their usernames. While adding the coder, the company will specify the starting and ending time of the interview.

The screenshot shows a 'Welcome' header with a menu icon and 'LOGOUT'. The main title is 'Add Participants'. Below it is a form with fields: 'Username' (input field), 'Start Date' (dd.mm.yyyy) and 'Start Time' (dropdown), 'End Date' (dd.mm.yyyy) and 'End Time' (dropdown). A blue 'ADD PARTICIPANT' button is at the bottom. Below this is a modal dialog with 'Username' and 'Name' input fields, and 'CANCEL' and 'SUBMIT' buttons.

## 9. See and Evaluate Interviews

### 9.1 See the List of Interview

Companies can list the interviews they held or are still holding. In this list they can view the position and duration of the interviews.

The screenshot shows a 'Welcome' header with a menu icon and 'LOGOUT'. The main title is 'Interviews'. Below it is a table with columns: 'Position', 'Duration', and 'See Candidates'. The table lists several interview positions with their respective durations and 'SEE CANDIDATES' buttons.

Position	Duration	See Candidates
Summer Intern 2022	0	<a href="#">SEE CANDIDATES</a>
Summer intern	0	<a href="#">SEE CANDIDATES</a>
Part time	0	<a href="#">SEE CANDIDATES</a>
Part time summer	0	<a href="#">SEE CANDIDATES</a>
Part time fall	0	<a href="#">SEE CANDIDATES</a>
Part time spring	56	<a href="#">SEE CANDIDATES</a>
Part time 2	45	<a href="#">SEE CANDIDATES</a>
Part time	50	<a href="#">SEE CANDIDATES</a>

## 9.2 See the List of Candidates

By clicking ‘See Candidates’ for an interview, companies can view the list of candidates for that interview.

### Candidates

Name	See results
Selbi	<button>SEE RESULTS</button>
Akin	<button>SEE RESULTS</button>

## 9.3 See Candidate’s Results

By clicking “See results” in Candidates list, the companies can view the results of the candidates.

Welcome Logout

Interview has not finished yet

<b>non coding</b> Non coding question <a href="#">SEE SUBMISSION</a>
<b>questio</b> Coding question <a href="#">SEE SUBMISSION</a>

## 9.4 Evaluate Candidate

When viewing the results of candidates, if the interview time has not passed yet for that candidate, then no 'Evaluate' button will be visible. Instead, Interview has not finished yet' text will be visible. If the interview time has passed for that candidate, then an 'Evaluate' interview button will be displayed, and the company can click it to choose the evaluation for that candidate.

The screenshot shows a user interface for interviewing candidates. At the top, there's a blue header bar with the text "Welcome" and "LOGOUT". Below the header, there are two sections: "non coding" and "questio". The "non coding" section contains the text "Non coding question" and "SEE SUBMISSION". To the right of the "non coding" section, the text "Interview has not finished yet" is displayed. The "questio" section contains the text "Coding question" and "SEE SUBMISSION".

The screenshot shows a user interface for interviewing candidates. At the top, there's a blue header bar with the text "Welcome" and "LOGOUT". Below the header, there's a section labeled "Q1 for interview" which contains the text "Coding question" and "SEE SUBMISSION". To the right of this section, there is a blue button labeled "EVALUATE".

The screenshot shows a user interface for interviewing candidates. A modal dialog box is open in the center of the screen. The dialog box has a white background and a thin black border. It contains the word "Gender" at the top. Below "Gender", there are six radio buttons arranged in two rows of three. The options are: "Strong No Hire", "No Hire", "Weak No Hire", "Weak Hire" (in the first row); and "Hire", "Strong Hire" (in the second row). At the bottom of the dialog box, there are two buttons: "CANCEL" on the left and "SUBMIT" on the right.