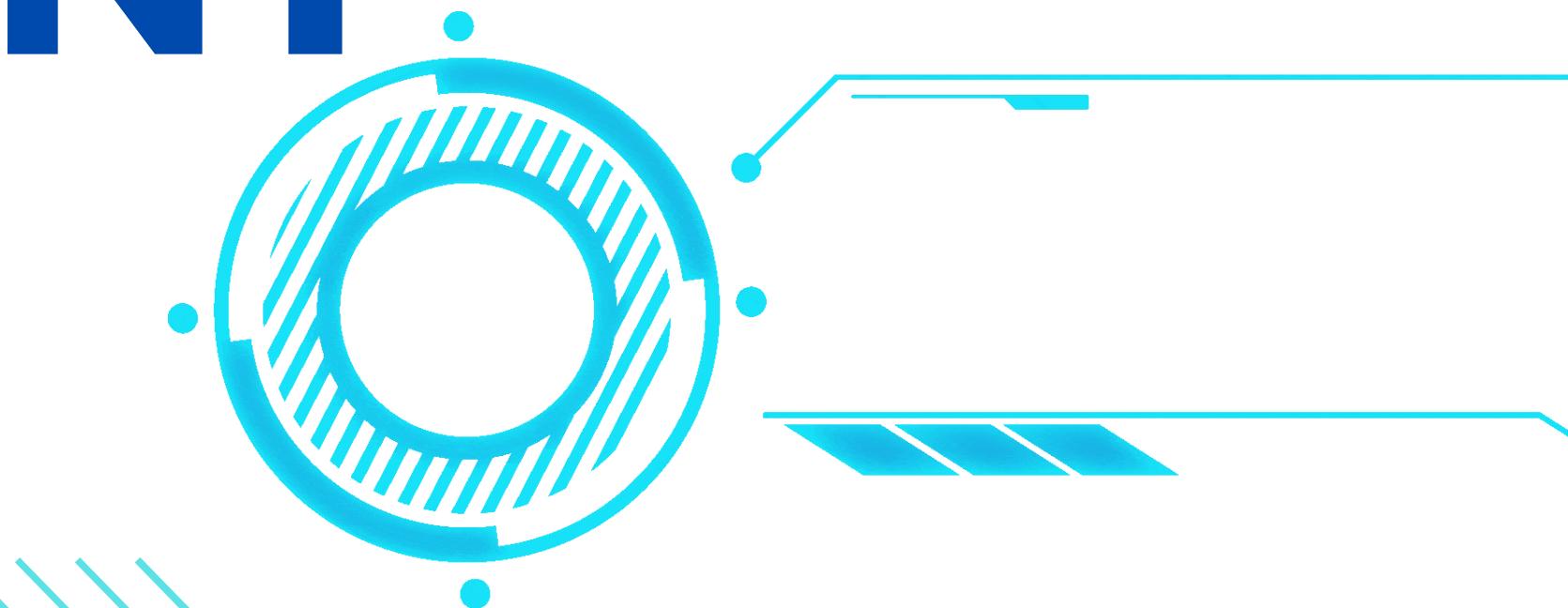




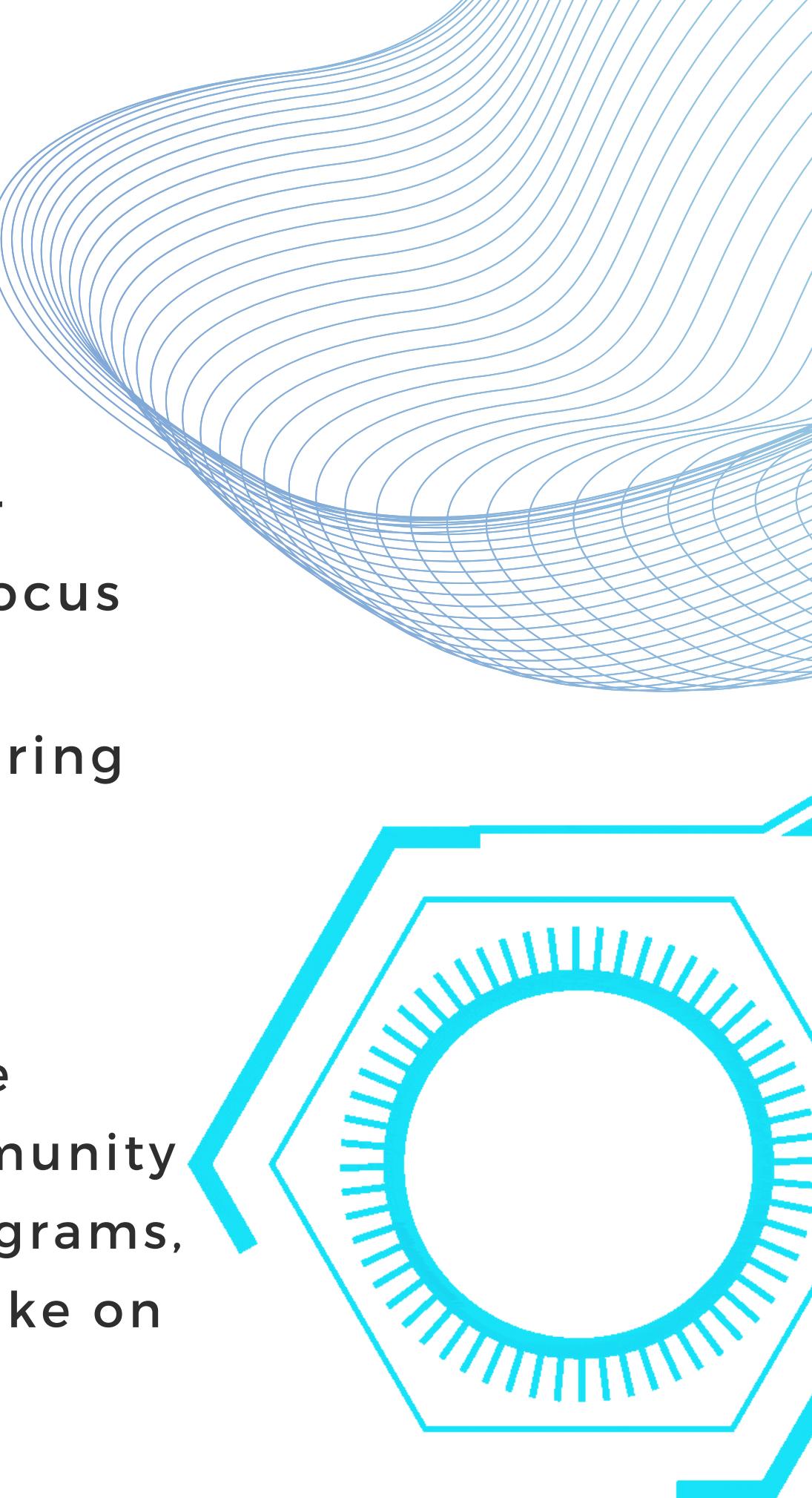
# JAVA DEVELOPMENT.



# ABOUT COMPANY

**CODSOFT** is a vibrant and diverse community that brings together individuals with similar objectives and ultimate goals. Our main focus is on creating opportunities that span various areas, including leadership development, learning, student engagement, and fostering shared interests.

We believe in the power of leadership and its ability to drive positive change. That's why we provide platforms and resources for our community members to develop their leadership skills. Through mentorship programs, workshops, and collaborative projects, we empower individuals to take on leadership roles and make a difference in their respective fields.



# INSTRUCTIONS

- Update your LinkedIn profiles
- For the Java Development internship, you will need to complete at least 3 tasks for successful completion of the internship.
- If you complete four tasks your chances get high to an extended internship with stipends. The code must be unique and don't copy the code.
- Maintain a separate GitHub repository(name as **CODSOFTJUNE** for all the tasks and share the link of the GitHub repo in the task submission form(it will be given later through email).
- You can refer to online resources such as Google Search and read tutorials. Watch videos(For Help).

# SUBMISSION

- A TASK SUBMISSION FORM will be shared later through email (Before 1 July 2023). Till then please continue your task.
- A video need to be created to showcase your work, a demo of your effort
- **For the Java Development internship, you will need to complete at least 3 tasks for successful completion of the internship.**
- **If you complete 4 tasks your chances get high to an extended internship with stipends. The code must be unique and don't copy the code.**
- The video can be hosted on LinkedIn for proof of your work and to build credibility among your peers. You can tag CODSOFT in such posts.
- Please add #codsoft in each of your task video postings on LinkedIn, Additionally, you can also add hashtags such as #internship #webdevelopment. for more reach and visibility.

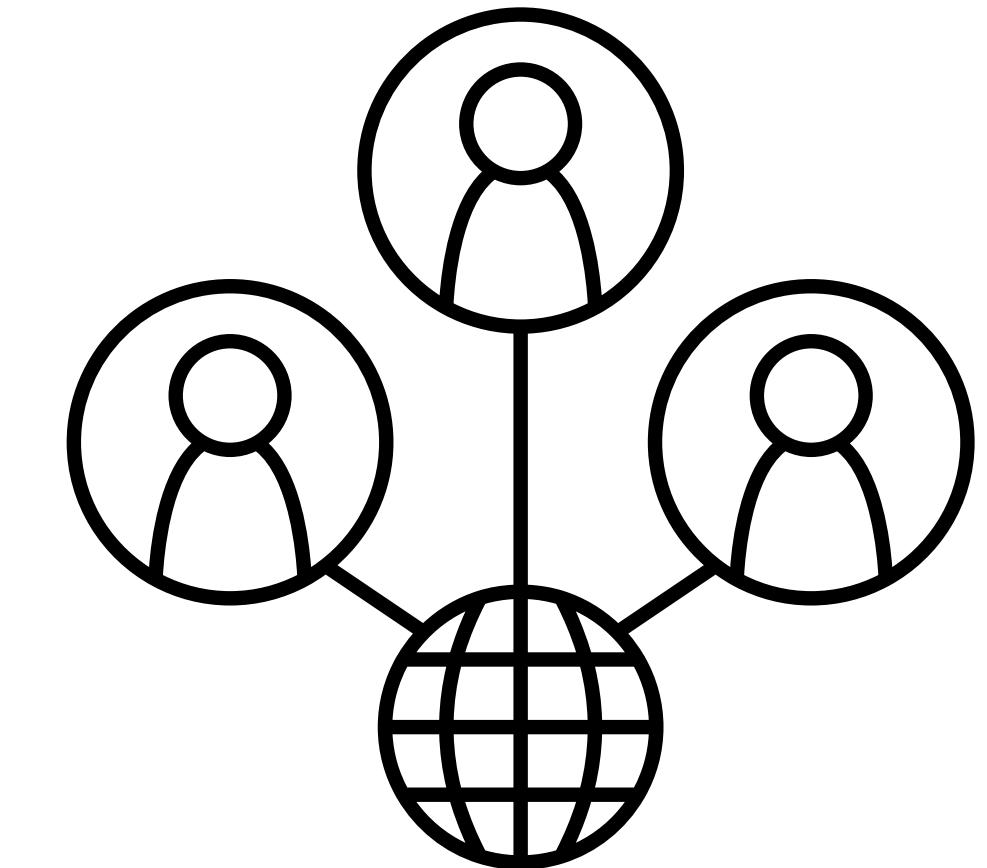
# ABOUT THE INTERNSHIP



**COMPLETION  
CERTIFICATE**



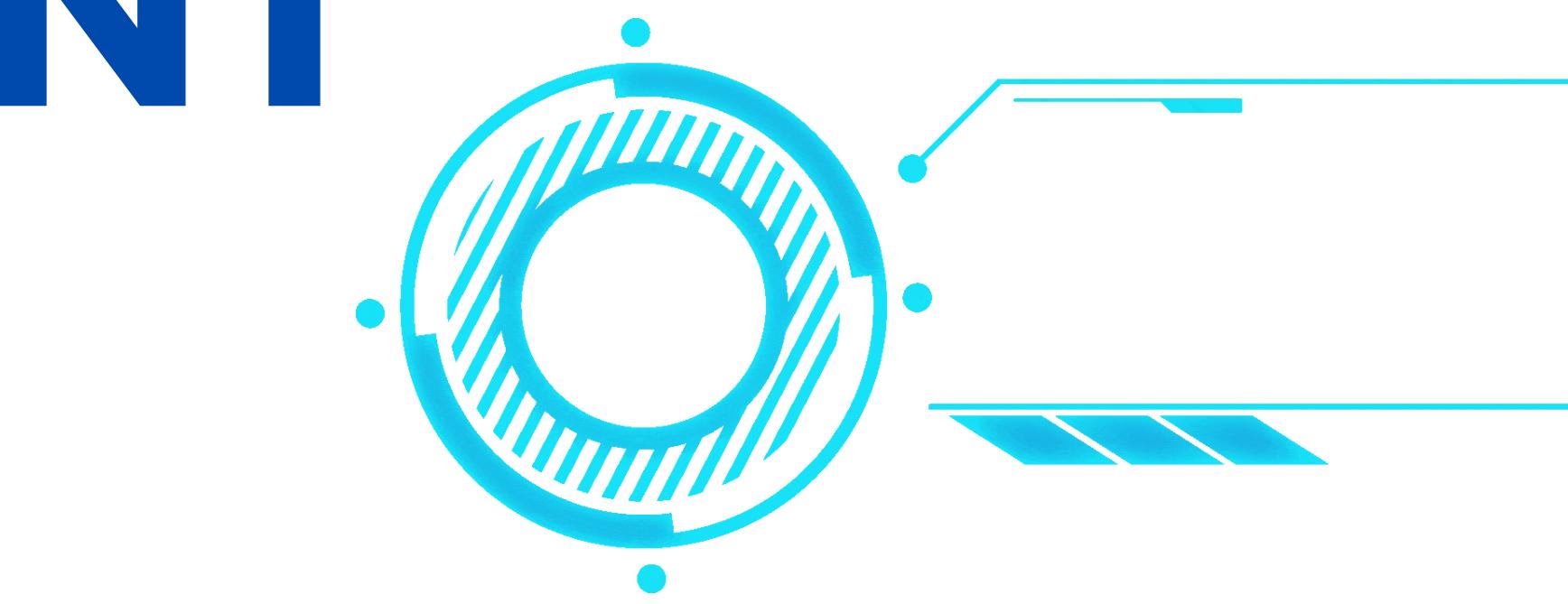
**PLACEMENT  
SUPPORT**



**NETWORK  
OPPORTUNITY**



# JAVA DEVELOPMENT



**For Java Development internship you will need to complete at least 3 tasks for successful completion of internship.**

# **TASK 1**

## **NUMBER GAME**

**FOR DEMO**  
**CLICK HERE**

1. Generate a random number within a specified range, such as 1 to 100.
2. Prompt the user to enter their guess for the generated number.
3. Compare the user's guess with the generated number and provide feedback on whether the guess is correct, too high, or too low.
4. Repeat steps 2 and 3 until the user guesses the correct number.

You can incorporate additional details as follows:

5. Limit the number of attempts the user has to guess the number.
6. Add the option for multiple rounds, allowing the user to play again.
7. Display the user's score, which can be based on the number of attempts taken or rounds won.

# **TASK 2**

## **WORD COUNTER**

**FOR DEMO**  
**CLICK HERE**

1. Prompt the user to enter a text or provide a file to count the words.
2. Read the input text or file and store it in a string.
3. Split the string into an array of words using space or punctuation as delimiters.
4. Initialize a counter variable to keep track of the number of words.
5. Iterate through the array of words and increment the counter for each word encountered.
6. Display the total count of words to the user.

You can further enhance the project by adding features such as:

7. Ignoring common words or stop words.
8. Providing statistics like the number of unique words or the frequency of each word.
9. Implementing input validation to handle empty inputs or file errors.
10. Adding a graphical user interface (GUI) for a more user-friendly experience.

# **TASK 3**

## **STUDENT MANAGEMENT SYSTEM**

**FOR DEMO**  
**CLICK HERE**

1. Create a Student class to represent individual students. Include attributes such as name, roll number, grade, and any other relevant details.
2. Implement a StudentManagementSystem class to manage the collection of students. Include methods to add a student, remove a student, search for a student, and display all students.
- 3 . Design the user interface for the Student Management System. This can be a console-based interface or a graphical user interface (GUI) using libraries like Swing or JavaFX.
4. Implement methods to read and write student data to a storage medium, such as a file or a database.
5. Allow users to interact with the Student Management System by providing options such as adding a new student, editing an existing student's information, searching for a student, displaying all students, and exiting the application.
6. Implement input validation to ensure that required fields are not left empty and that the student data is in the correct format

# **TASK 4**

## **ATM INTERFACE**

**FOR DEMO**  
**CLICK HERE**

1. Create a class to represent the ATM machine.
2. Design the user interface for the ATM, including options such as withdrawing, depositing, and checking the balance.
3. Implement methods for each option, such as withdraw(amount), deposit(amount), and checkBalance().
4. Create a class to represent the user's bank account, which stores the account balance.
5. Connect the ATM class with the user's bank account class to access and modify the account balance.
6. Validate user input to ensure it is within acceptable limits (e.g., sufficient balance for withdrawals).
7. Display appropriate messages to the user based on their chosen options and the success or failure of their transactions.

# **TASK 5**

## **ADDRESS BOOK SYSTEM**

**FOR DEMO**  
**CLICK HERE**

1. Create a Contact class to represent individual contacts. Include attributes such as name, phone number, email address, and any other relevant details.
  
2. Implement an AddressBook class to manage the collection of contacts. Include methods to add a contact, remove a contact, search for a contact, and display all contacts.
  
3. Design the user interface for the Address Book System. This can be a console-based interface or a graphical user interface (GUI) using libraries like Swing or JavaFX.
  
4. Implement methods to read and write contact data to a storage medium, such as a file or a database.
  
5. Allow users to interact with the Address Book System by providing options such as adding a new contact, editing an existing contact's information, searching for a contact, displaying all contacts, and exiting the application.
  
6. Implement input validation to ensure that required fields are not left empty and that the contact data is in the correct format.

# **ASK US FOR HELP!**

- **THE PURPOSE OF THIS INTERNSHIP IS TO LEARN AND GROW**
- **We have no desire to dictate to you. It is entirely up to you whether you seek guidance or not.**
- **The given tasks may seem very easy or very difficult. We expect you to approach the tasks with professional diligence and give them the attention they deserve."**

# GET SOCIAL WITH US



**www.codsoft.in**



**contact@codsoft.in**



**CodSoft**