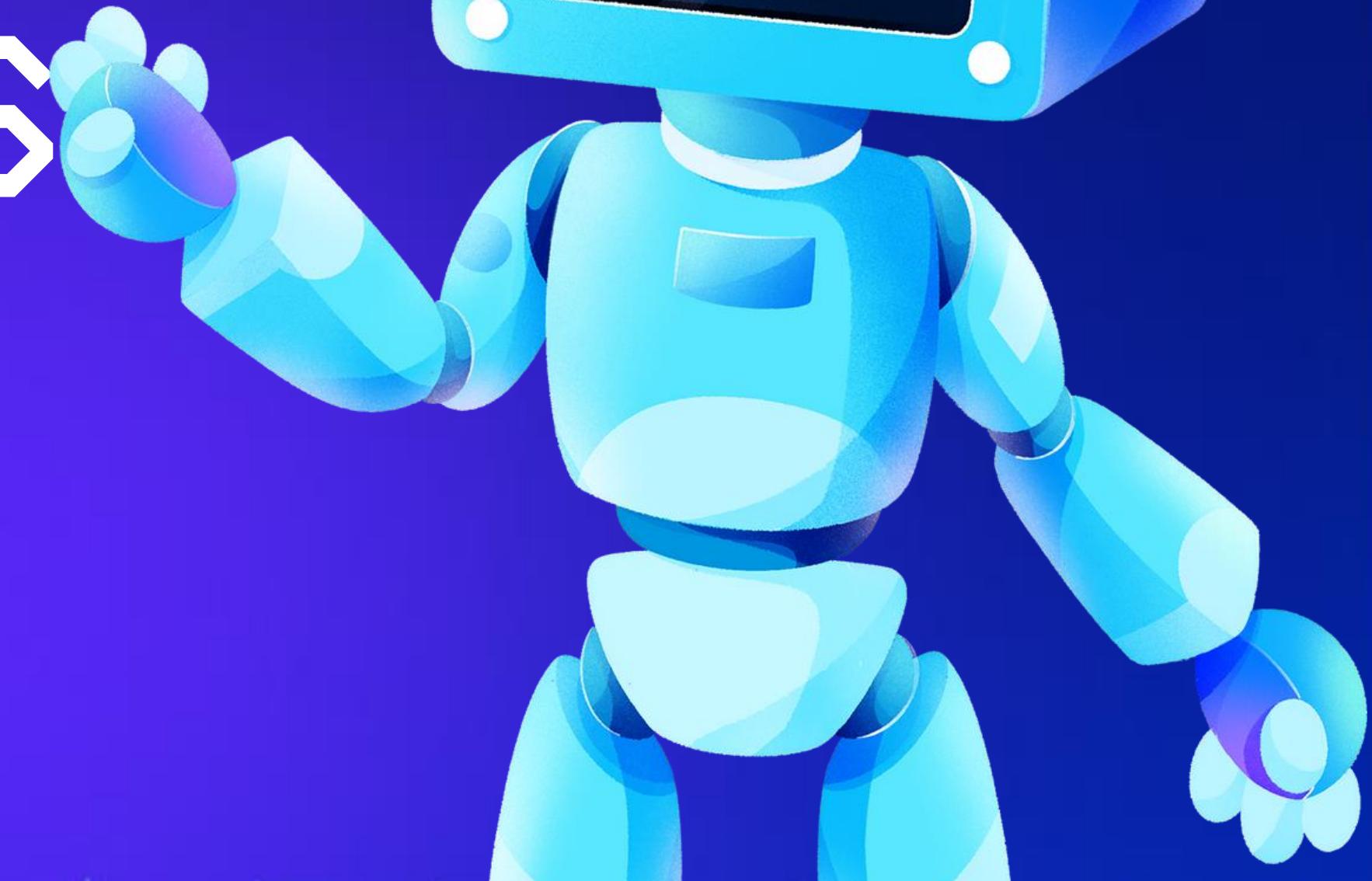
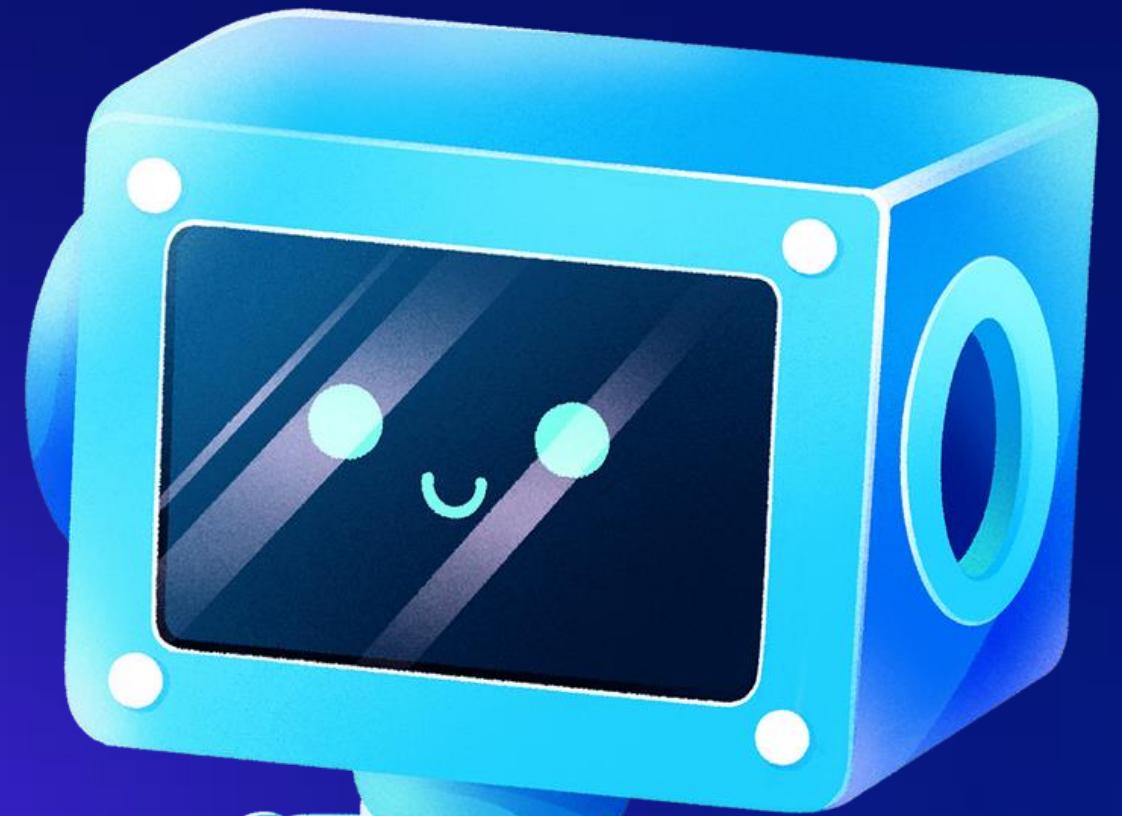


JAVA DEVELOPMENT PROJECTS

By Encryptix



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.

Maintain a separate GitHub repository(name as **Encryptix**) 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 . 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.**

The video can be hosted on LinkedIn for proof of your work and to build credibility among your peers.
You can tag @ENCRYPTIX in such posts.

Please add #encryptix 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 the JAVA Development internship, you will need to complete at least 3 tasks for successful completion of the internship.



TASK 1

NUMBER GAME

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

STUDENT GRADE CALCULATOR

Input: Take marks obtained (out of 100) in each subject.

Calculate Total Marks: Sum up the marks obtained in all subjects.

Calculate Average Percentage: Divide the total marks by the total number of subjects to get the average percentage.

Grade Calculation: Assign grades based on the average percentage achieved.

Display Results: Show the total marks, average percentage, and the corresponding grade to the user

TASK 3

ATM INTERFACE

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 4

CURRENCY CONVERTER:

Currency Selection: Allow the user to choose the base currency and the target currency.

Currency Rates: Fetch real-time exchange rates from a reliable API.

Amount Input: Take input from the user for the amount they want to convert.

Currency Conversion: Convert the input amount from the base currency to the target currency using the fetched exchange rate.

Display Result: Show the converted amount and the target currency symbol to the user.

STUDENT MANAGEMENT SYSTEM

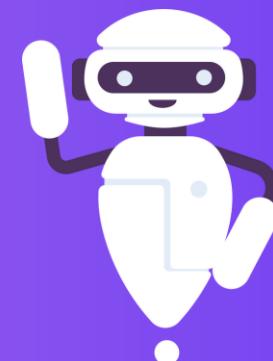
1. Create a Student class to represent individual students. Include attributes such as name, roll number, grade, and any other relevant details.
2. Implement a StudentManagementSystem class to manage the collection of students. Include methods to add a student, remove a student, search for a student, 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

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.encryptix.in



contact@encryptix.in



[Encryptix](#)

