BANSIRA
The Superapp, Always There For You

PRACTICAL

The technical task is divided into two parts: the first part must be completed using Java, and the second part must be completed using Node.js with TypeScript (preferably using NestJS).

Java Task (Part 1)

Implement Library Management System

A library has multiple departments and each departments has multiple books, but same book does not exist in another department

You need to develop a Library Management system and Leadership Board.

Requirements:

- **Classes:**
  1. Book: Represent a book with attributes like title, author, ISBN, genre, publication year, Departments, and availability (boolean). Implement appropriate getters and setters.
  2. Library: Manage a collection of books in multiple departments. It should include methods:
  3. addBook(Book book): Adds a book to the library's Departments collection (check for duplicates based on ISBN).
  4. removeBook(String ISBN): Removes a book by ISBN.
  5. findBookByTitle(String title): Returns a list of books matching the title (case-insensitive search).
  6. findBookByAuthor(String author): Returns a list of books by the author (case-insensitive search).
  7. listAllBooks(): Returns a list of all books in the library.
  8. listAvailableBooks(): Returns a list of available books.
  9. LibraryMenu: (Optional) Create a simple text-based menu for library operations (add, remove, search, list, exit).

- **Unit Tests:**
  Write unit tests for all methods in the Book and Library classes using a testing framework like JUnit (encouraged). This demonstrates the candidate's understanding of test-driven development (TDD) principles.

- **Submission:**
  Package your Java source code (Book.java, Library.java, LibraryMenu.java (optional), unit test classes with *.java files) and any relevant test runner libraries (e.g., JUnit JAR) into a ZIP file.

NodeJs Task (Part 2)

Implement Leadership/Ranking Board

You need to implement two Leadership/Ranking Board

1. Popular department - It will show last week winner and top 5 department ranking, department ranking will update every day
2. Popular Books - Here librarian has to show books ranking for
(i) Weekly popular
(ii) Monthly Popular
(iii) Today Trending (will update every one hour)

Considerations:

1. Books will be removed after same book remains in least read/downloaded for consecutive 2 weeks.
2. Ranking will be calculated on basis of downloaded books

Recommendations:

1. Consider using an IDE like IntelliJ IDEA or Eclipse for code completion, refactoring, and debugging.
2. Break down the task into smaller, manageable steps.
3. Start with unit tests to guide your development process (TDD approach).
4. Write clear and concise code documentation (comments).
5. Test your application thoroughly with various input scenarios.
6. Feel free to add your own creative ideas to enhance the system.

Evaluation Criteria:

- **Code Quality:**
  Clear and well-commented code that adheres to Java coding conventions.
  Use of appropriate data structures (e.g., ArrayList for book collections).
  Proper exception handling.

- **Unit Testing:**
  Comprehensive unit test coverage for all methods.
  Demonstrates understanding of testing practices.

- **Optional Extra Credit:**
  Implement the LibraryMenu class for a user-interactive experience.
  Explore design patterns for a more robust library system.