ShadowFox

LEARN • CREATE • LEAD

Java Development Internship Task List

Internship Pre-requisites before starting your tasks:

1.LinkedIn Profile Update: Ensure that your LinkedIn profile is updated to reflect your technical skills, and update your experience section to include **"ShadowFox Java Development Intern."**

2.LinkedIn Post: It's not mandatory to post the offer letter on LinkedIn, but if you wish to receive **extra swags** at the end of the internship, you can post your offer letter and **tag us** to receive assured swags.

3.GitHub Repository: You are required to create a separate GitHub repository named **"ShadowFox"** for all tasks. You can use any IDE to write your code and upload it to the respective repository.

4. Completion of Tasks: Complete the required tasks as specified in this Task List.

5. Proof of Work: At ShadowFox, we value **Proof of Work** (POW). You are required to post a video explanation of your respective tasks in LinkedIn. Screenshots must be submitted during your task submission.

After completing all the above steps, proceed with your task completion.Kindly note that all the details and screenshots you submit will be thoroughly verified.

Task Level (Beginner): (Mandatory to do any 2 tasks out of 3)

1)Enhanced Console-based Calculator:

Develop a console-based calculator that includes basic arithmetic operations (addition, subtraction, multiplication, division) along with additional functionalities such as scientific calculations (e.g., square root, exponentiation) and unit conversions (e.g., temperature, currency).
Key Learning: Basic Java syntax, methods, error handling, and additional mathematical operations.

2)Simple Contact Management System:

Create a simple command-line contact management system using Java. Allow users to add, view, update, and delete contacts. Store contact information like name, phone number, and email address in memory (no database required).
Key Learning: Basic CRUD operations, user input/output, array or ArrayList data structures.

3)Student Information System with GUI (JavaFX/Swing):

Create a Student Information System with a Graphical User Interface (GUI) using JavaFX or Swing. The system should allow users to add, display, update, and delete student records visually.
Key Learning: GUI development, event handling, basic CRUD operations in a graphical environment.

Task Level (Intermediate): (Mandatory to do any 2 tasks out of 3)

1)Bank Account Management System with Unit Testing (JUnit):

Extend the Bank Account Management System to include unit testing using JUnit. Write test cases to validate functionalities such as deposit, withdrawal, balance inquiry, and transaction history.
Key Learning: Unit testing, test-driven development, software testing principles.

2)Inventory Management System with Basic GUI:

Develop an Inventory Management System with a basic Graphical User Interface (GUI) using JavaFX or Swing. Implement functionalities like adding, updating, and deleting inventory items.
Key Learning: GUI design, event handling, basic CRUD operations in a graphical environment.

3)Library Management System with Data Persistence (SQLite):

Enhance the Library Management System with features like user accounts, book recommendations, and integration with an external book information API (optional). Use SQLite for data persistence instead of a full-fledged database.
Key Learning: Data persistence with SQLite, API integration (optional), user management.

Task Level (Advanced): (Do any 1 task out of 3)

1)Online Banking System with RESTful API (Spring Boot):

Develop an Online Banking System with a RESTful API using Spring Boot. Implement features such as account creation, funds transfer, transaction history, and authentication.
Key Learning: RESTful API development, Spring Boot framework, secure web services.

2)Real-time Chat Application with Java Socket Programming:

Build a real-time chat application using Java Socket Programming. Allow multiple users to join chat rooms, send messages, and receive instant updates.
Key Learning: Networking fundamentals, socket programming, real-time communication.

3)E-commerce Website Backend with Microservices Architecture:

Build a backend system for an e-commerce website using a microservices architecture. Break down functionalities such as user authentication, product management, and order processing into independent microservices.
Key Learning: Microservices architecture, distributed systems, scalability, modularity.

Resources: https://roadmap.sh/java