



**Internship Provided By:
DataInn Technologies**

Name: Yunus Aslam Bagewadi
Class: T.Y.B.Sc.Computer Science
Div: A
Roll No.: 210

About DataInn Technologies:

DataInn Technologies is an AI-focused IT consulting firm that helps organizations design, build, and scale production-ready AI solutions. The company specializes in AI agents, data platforms, and decision intelligence, enabling businesses to move from experimentation to real-world impact.

Industries Served:

- Finance & FinTech
- Healthcare & Life Sciences
- Legal & Professional Services
- Retail & E-commerce
- Manufacturing & Enterprise Operations

Core Expertise:

- Artificial Intelligence & Machine Learning
- Large Language Models (LLMs) & AI Agents
- Data Engineering & Analytics
- Cloud & Enterprise AI Architecture

Domain: Advance Java

Mode: Remote

Internship Period: 18-November-2025 To 18-February-2025

Technologies Actively Working On:

- Java
- JDBC
- PostgreSQL
- React JS
- JPA
- SDLC Documentation
- Git/GitHub
- JSON APIs
- JWT Authentication...

Tasks Assigned:

Week-1-Tasks:

- Understanding how to **create a simple database, tables, insert data, and write basic SQL queries.**
- Learn how to connect Java to PostgreSQL and perform very simple **CRUD operations.**
- GitHub Setup & Push: **Version control** and code publishing.

- Understanding and implementing **Object Oriented Programming** concepts.
- Creating modules like BankAccount, StudentMGMT, EmployeeMGMT, etc.
- Implementing **JDBC CRUD operations** via OOPs concepts.

Week-2-Tasks:

- **Implementing Collection built-in data structures such as:**

- ArrayList<> ◦ HashSet<>
- TreeSet<> ◦ LinkedList<>
- Map<> ◦ HashMap<>
- TreeMap<> ◦ LinkedHashMap<>

- **And implementing problems based on above Collections, such as:**

- Browser Tab History Behavior ◦ Exam Submission Behavior

- Understanding which Collection data structure is efficient for a given scenario / problem.

- **Implementing Iteration Classes and Methods, which included concepts like:**

- Iterator<> ◦ ListIterator<> ◦ ForEach Iterator
- Fail-Fast Iteration ◦ Fail-Safe Iteration

Week-3-Tasks:

◦ Understanding Types of **built-in exceptions** and **user-defined exceptions**, when are they thrown in order to ensure smooth execution of a program / software.

◦ **Creating custom user-defined exceptions such as:**

- InsufficientBalanceException
- InvalidAgeException
- InvalidLoginException

◦ **And using these exception with modules below:**

- BankAccount
- RatedRMovie
- LoginValidation

◦ **Implementing File-Handling Reader Classes and Methods such as:**

- File
- FileReader
- BufferedReader

◦ **Implementing File-Handling Writer Classes and Methods such as:**

- FileWriter
- BufferedWriter

◦ **And using above File-Handling concepts to implement behaviors like:**

- Log Audit Trails
- Filtering .csv/.txt Files
- Formatting Output

Week-4-Tasks:

- Understanding how **SDLC documentation is done and implemented**. And creating a hypothetical project documentation for "SmartClinic 2.0" and **understanding concepts such as:**
 - Feasibility Study ◦ Software Requirement Specification (SRS)
 - System Study ◦ Time & Resource Estimation
 - Workflow Diagrams ◦ Functional Requirements (FR)
- Attempt to create a **Real Industry-Level documentation** for a project called "School Management System" in which assignments, exam time-tables, notifications, attendance, etc. will be done and digitally recorded. **Documentation included were:**
 - Business Requirement Document (BRD) ◦ Requirement Documentation
 - Software Requirements Specification (SRS) ◦ Risk & Mitigation Analysis
 - ER Diagram ◦ Workflow Diagram ◦ UI Wireframes
- And focusing on creating detailed **Module Based Functional Requirements (FR) Documentation** which included describing module specific users, inputs / outputs of that module, and by which user is the module most used by.

Week-5-Tasks:

- Build a basic **REST API** using **Spring Boot** that performs CRUD operations on a database table.
- Created **Entity (Class)**, **JPA Repository**, **Service Layer** and **Rest APIs Controller** with standard methods such as **POST**, **GET**, **PUT**, **DELETE**. Used Spring Boot Annotations accordingly.
- Database Configured and APIs endpoints testing using **POSTMAN**.
- Advance JPA DB schemas fetching using custom queries.
- **Pagination and Sorting** of the data being fetched using **Pageable** interface.
- Implementation of **Liquibase (Github of Databases)**, tracking every DB schemas change done efficiently and migrating the DB changes in the next deployment of the application.
- **Unit Testing** Service Layer (Business Logic) using **Mockito** and Controller Layer (Rest APIs) using **@WebMvcTest** Spring Boot test annotation.
- **Integration Testing** of Repository Layer (DB Fetching) and Liquibase using **@DataJpaTest** annotation which was done in **TestContainers (Docker)**.

THANK YOU FOR YOUR TIME

