MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

Final Year Internship Report

on

Academic Intern at Persistent Systems

Submitted By:

Anushka Singh Tomar

0901CS181016

Faculty Mentor:

Prof. Khushboo Agarwal

Professor, CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE

GWALIOR - 474005 (MP) est. 1957

MAY-JUNE 2022

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

Academic Intern at Persistent Systems

A final year internship report submitted in partial fulfilment of the requirement for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

Submitted by:

Anushka Singh Tomar

0901CS181016

Internship Faculty Mentor:

Jayati Munot, Associate Executive, Learning and Development,

Persistent Systems

Submitted to:

Prof. Khushboo Agarwal

Assistant Professor CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE

GWALIOR - 474005 (MP) est. 1957

MAY-JUNE 2022

Internship Certificate Received from Industry/Company

Persistent

PSL/HR/Cert-Add/2022
May 18, 2022

TO WHOMSOEVER IT MAY CONCERN

This is to certify that **Ms. Anushka Tomar** (Employee Code **46282**) is employed with us since **12 January 2022**. Her designation is **Intern**.

According to the office records, her local and permanent addresses are :

Local Address	**Permanent Address**
F-67 Kotra Sultanabad Government Quarters, Bhopal - 462003 Madhya Pradesh -India	F-67 Kotra Sultanabad Government Quarters, Bhopal - 462003 Madhya Pradesh - India

This certificate is being issued on her request as a proof of employment and residence For verification purpose.

For Persistent Systems Ltd.

Manisha Tapaswi
Senior General Manager - Human Resources

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

CERTIFICATE

This is certified that **Anushka Singh Tomar**(0901CS181016) has submitted the Internship report titled **Software Development Engineer Intern** of the work he has done under the mentorship of **Prof. KhushbooAgarwal**, in partial fulfilment of the requirement for the award of degree of Bachelor of Technology in Computer Science and Engineering from Madhav Institute of Technology and Science, Gwalior.

Prof. KhushbooAgarwal

Assistant Professor

Computer Science and Engineering

27/05/21

Dr. Manish Dixit

Professor and Head,

Computer Science and Engineering

Dr. Manish Dixit
Professor & HOD
Department of CSE
M.I.T.S. Gwalior

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

DECLARATION

I hereby declare that the work being presented in this Internship report, for the partial fulfilment of requirement for the award of the degree of Bachelor of Technology in CSE at Madhav Institute of Technology & Science, Gwalior is an authenticated and original record of my work under the mentorship of **Prof. KhushbooAgarwal,Assistant Professor**, Department of **CSE**.

I declare that I have not submitted the matter embodied in this report for the award of any degree or diploma anywhere else.

Anushka Singh Tomar

0901CS181016,

IV Year,

Computer Science and Engineering

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

ACKNOWLEDGEMENT

The full semester internship has proved to be pivotal to my career. I am thankful to my institute, **Madhav Institute of Technology and Science** to allow me to continue my disciplinary/interdisciplinary internship as a curriculum requirement, under the provisions of the Flexible Curriculum Scheme (based on the AICTE Model Curriculum 2018), approved by the Academic Council of the institute. I extend my gratitude to the Director of the institute, **Dr. R. K. Pandit** and Dean Academics, **Dr. Manjaree Pandit** for this.

I would sincerely like to thank my department, **Department of Computer Science and Engineering, for allowing** me to explore this internship. I humbly thank **Dr. Manish Dixit**, Professor and Head, Department of Computer Science and Engineering, for his continued support during the course of this engagement, which eased the process and formalities involved.

I am sincerely thankful to my faculty mentors. I am grateful to the guidance of, **Prof. Khushboo Agarwal, Professor**, Department of Computer Science and Engineering, for her continued support and close mentoring throughout the internship. I am also very thankful to the faculty and staff of the department.

Anushka Singh Tomar

0901CS181016,

IV Year,

Computer Science and Engineering

ABSTRACT

In the present era technology has been the main concern because it is the one which connect one to other and has changed the lives of our and for this, we need software and applications to run on platform. Here I have assigned as Software Engineer Intern. My internship is basically divided into two phases phase 1 and phase 2 and right now my phase-1 training is going on in which I have learnt about different modules which form the key components to build any software and then in phase-2 training I will undergo the process to learn advance module and then the project will be allocate.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviation	Description
1.DBMS	Database Management System
2.RDBMS	Relational Database Management System
3.SQL	Structured Query Language
4.POJO	Plain Old Java Object
5.JSP	Java Server Pages
6.JPA	Java Persistence API
7.ORM	Object Relational Mapping
8.POM	Project Object Model

Chapter 1: Internship Overview

1.1 Introduction

Persistent Systems is a trusted digital engineering and enterprise modernization partner, combining deep technical expertise and industry experience to help our clients anticipate what's next and answer questions before they're asked. The offerings and proven solutions create unique competitive advantage for the clients by giving them the power to **see beyond and rise above**.

In my Internship, I have been offered a 6 month Program starting from January 2022 till July 2022. The Internship has been divided intio several technologies comprising of Frontend technologies like React.Js, Angular etc , Backend Technologies like Core Java , Java Fullstack, C++, .Net etc. and cloud technologies. As I mentioned, I have been assigned the Core Java Track in which I have been trained about Git and Github, Object Oriented System Design using UML and Core Java(1.8), Apache Maven (3.8.4), Spring 5, Spring Security and Spring Boot, HTML5, JavaScript(ES6).

1.2 Objective and Scope

The objective of the Internship Program is to create future Software Engineers who will be joining the Organization after successfully completing all the required criteria of the Company Policy.

1.3 Internship Features

The Interns are trained on particular technology tracks assigned to them and after completing the training and modules they have to go through GEMS Assessments.

On the basis of the performance in these assessments they are evaluated and sent to further trainings.

After completing the whole training in particular technology track, Interns are assigned to the Business Units according to the requirement of the Company.

Chapter 2 : Git (Version Control System) :

2.1 Introduction to Git :

A free and open source distributed version control system, i.e. Git is made to do everything from tiny to larger projects as well with efficiency.

It is better than SCM tools like Subversion, CVS, Perforce,

Git comes with built-in GUI tools (git-gui, gitk).

2.2 Benefits of Git :

•Almost all Git operations require local files and resources to work.

• Three states – Git working directory, staging area, and repository.

• Everything is done offline.

• There is no central authority.

• Changes can be shared without a server.

• Complete project history on your local hard drive. Most operations are almost instantaneous.

• If you don't have server or VPN access, you don't have to wait for access as everything is available locally unless you get it from a friend (peer).

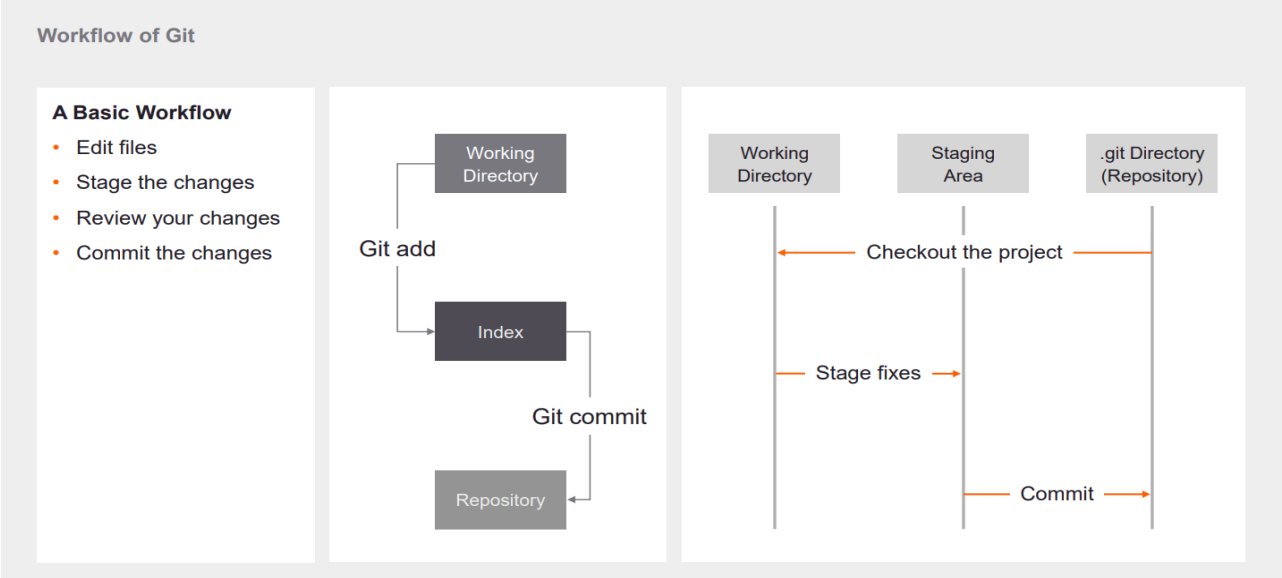


Fig 2.2 Workflow of Git

2.3 Working with Git Local :

The actions that Git can perform locally are:

• Create local repository, add files, commit changes

• View logs and differences

• Deploy changes as multiple changes

• Delete and rename files o Ignore files

• Undo / rechange changes Trial Local Copy and Repository

• Clean up your working copy

2.4 Git Commands:

git version

Outputs the version of the Git suite that the Git program was based on.

git –help

Prints a summary and list of the most commonly used commands. If you specify all or the option, all available commands are printed. If the Git command is named, this option opens the man page for that command.

2.4.1 Git Local :

(I) Git init:

git init

This page describes the git init command in detail. At the end of this page you will be informed about the core and extensions of git init. This search includes

(iii) Git clone:

git clone

[url] gets the complete repository from the location hosted via the URL

(ii) Git config:

git config global user.name ""

Set a name that can be identified as a credit when checking the version history Directory>

Change to for the next release Commit To modify a particular file, replace directory> with .

(v) Git commit:

git commit m * Commits the provided snapshot, but uses as the commit message instead of launching the text editor

(vi)) Git status:

git status

Lists deployed, undeployed, and untracked files

(vii) Git log:

• git log –n 3 → shows only 3 commits

• gitl og–oneline → Combine each commit into one line

• git log author = "" → Search for commits by a particular author. Arguments can be simple strings or regular expressions

• git log grep = "" → Search for commits that contain a commit message that starts with a match A simple string or a regular expression.

2.4.2 Working with Git Remote (Github):

• GitHub is a way for people to build software

• Developers with millions of communities can discover and contribute projects using powerful co-development workflows.

• GitHub is a code hosting platform for version control and collaboration.

• Allows you and others to collaborate on a project from anywhere.

• GitHub has become the world's largest online repository for collaboration.

command:

git remote add → Create a new connection to the remote repository. After adding the remote control, you can use as a shortcut for in other commands.

git fetch → Push the branch to with the required commits and objects. If the named branch does not exist, create it in the remote repository.

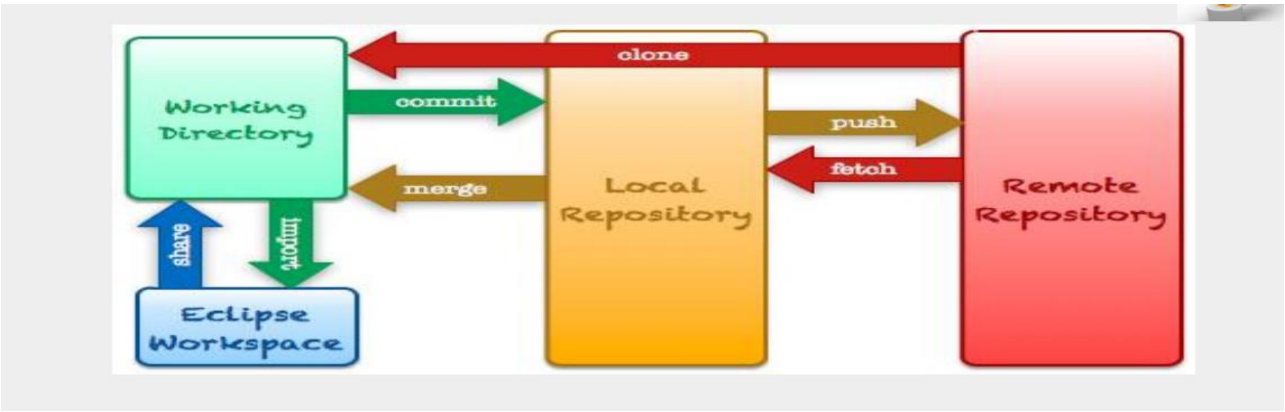


Fig 2.4.2 Git Lifecycle

CHAPTER 3: OOPS

3.1 UML Diagram

The UML diagram is a UML (Unified Modeling Language) -based diagram that visually represents, changes, maintains, or documents the system along with its key actors, roles, actions, artifacts, or classes. The purpose is to get. Information about the system. You can also simplify complex software design and implement OOP as a general concept.

3.2 UML Class Diagram

UML Class Diagram is a graphic representation for building and visualizing object-oriented systems. A Unified Modeling Language (UML) class diagram is a type of static structural diagram that shows the structure of a system as follows: Between

• class,

• its attributes,

• operation (or method),

• and related objects.

3.3 UML class notation

class represents the concept of encapsulating states (attributes) and actions (operations). Every attribute has a type. All operations have a signature. All you need is the class name.

Fig 3.3 Class Signature in UML

3.3.1 Class Name:

• The class name is displayed in the first section.

3.3.2 Class Attributes:

• Attributes are displayed in the second section.

• Attribute types appear after colons.

• Attributes are mapped to member variables (data members) in code.

3.3.3 Class operations (methods):

• Operations are shown in the third section. This is the service provided by the class.

• The return type of a method is specified after a colon at the end of the method signature.

• The return type of a method parameter is specified by the parameter name followed by a colon. Mapping actions to class methods in code

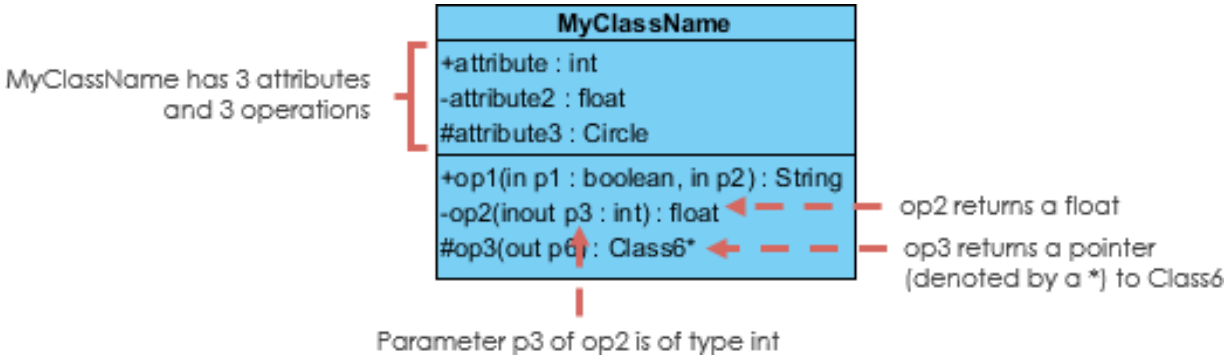


Fig 3.3.3 Class Notation

3.3.4 Class Visibility

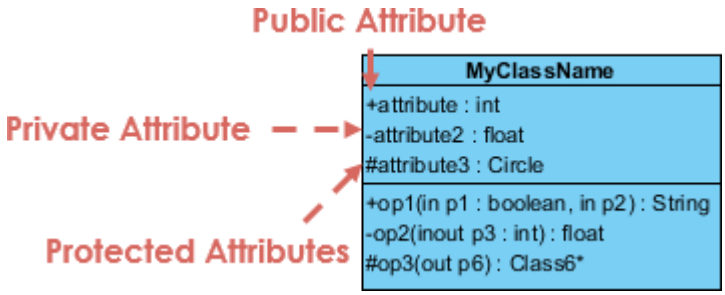
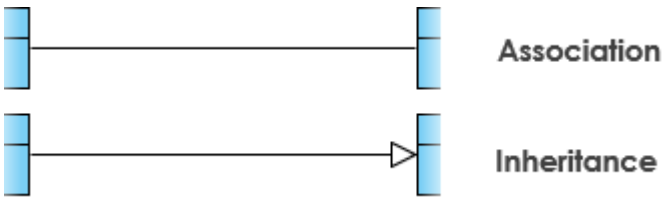


Fig 3.3.4 Class Visibility

- \+ indicates public attributes or operations
- \- indicates private attributes or operations
- # indicates protected attributes or operations
-

3.4 Relationships Between Classes

UML is more than just a picture. When used correctly, UML communicates exactly how to implement code in a diagram. When interpreted correctly, the implemented code correctly reflects the designer's intent.



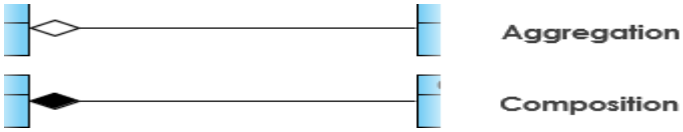


Fig 3.4 Relationship between classes

3.4.1 Inheritance (or generalization):

A generalization is a taxonomic relationship between a more general classifier and a more specific classifier. Each instance of a particular classifier is also an indirect instance of a generic classifier. Therefore, some classifiers inherit the functionality of more general classifiers.

• Represents an "is-a" relationship.

• Abstract class names are italicized.

.

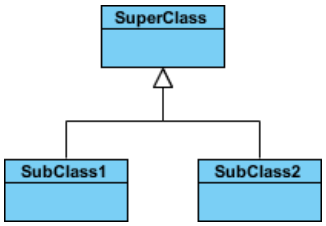


Fig 3.4.1 Inheritance

3.4.2 Associations

Associations are relationships between classes in a UML class diagram. It is indicated by a solid line between classes. Associations are usually named as verbs or verb phrases that reflect real-world domains.

Cardinality: The cardinality is expressed as

• One-to-one

• One-to-many

• Many-to-many

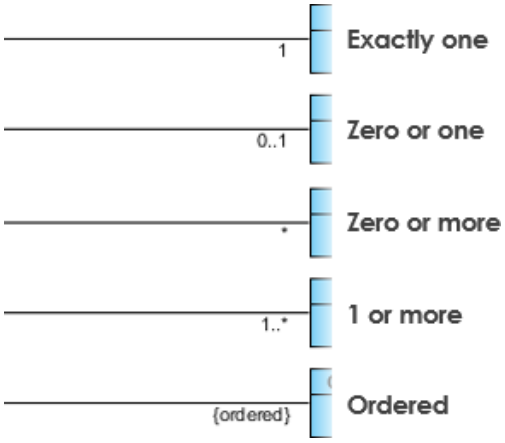


Fig 3.4.2 Association

3.4.3 Aggregation

A special type of association.

• Represents "Part" of a relationship.

• Class 2 is part of Class 1.

• Many instances of class 2 (marked with *) can be associated with class 1.

• Class1 and Class2 objects have different lifetimes.

3.4.4 Composition

• A special type of aggregate whose parts are destroyed when the whole is destroyed.

• Class-2 objects live and die with class-1.

• Class 2 cannot operate on its own.

Chapter 4 SQL

4.1 Database

A database is an organized collection of related data, usually stored on a hard drive and made available to many concurrent users. A database management system (DBMS) is a program for working with databases.

Earlier developers used file system to manage their data, but there are many problems in using file system to manage data. Therefore, the file system is replaced by a database management system.

Advantages of the DBMS:

• Reduction of data redundancy

• Reduced update errors and increased consistency

• Improving data integrity and application independence

• Improved data access for users using host and query languages.

• Greater data security

• Reduce the cost of entering, storing and retrieving data

• Facilitate the development of new application programs

As with all products, DBMS has its disadvantages, so the disadvantages of DBMS are:

• The design of database systems is complex, complex and time-consuming.

• Significant start-up costs for hardware and software.

• Database corruption is not good for any connected applications.

• High conversion costs when changing from a file system to a database system.

4.2 Entity-Relationship Model of objects

The most common database management system in this scenario is a relational database management system called RDMS.

Object: An object is a real thing or an object that can be distinguished from other objects by the value of a property.

Property: Unit that describes the properties of an object.

• Simple properties.

• Complex Properties

• Delete Properties

• Multi-Valued Properties

• Derived Properties

• Stored Properties

4.2.1 Relationship:

A relationship between two or more objects in different object sets. There are basically three types:

• **One-to-one relationship**: Each row in one table is related to a row in another table. For example, a studen

t might have an address. The address belongs to the student.

• **One to Many Relationships** : Each row in one table references multiple rows in another table. For example, a team consists of several players but a player only belongs to his team(one team).

• **Many-to-many relationships**: Each row in each table relates to multiple rows in another table. Each object can be associated with multiple instances of other objects. For example, a project may contain multiple employees, and employees may belong to more than one project.

4.3 RDBMS:-

Data in a DBMS is stored in database objects called tables. This table is a set of related records and consists of many columns and rows.

4.3.1 Some terms

• **Table**. A table is the basic storage structure of an RDBMS, consisting of one or more columns and zero or more rows.

• **Column**: A column indicates the data type of the table. Also called object property.

• **Field. Margins** are the dots that connect the rows and columns of a table. If a field contains no data, it is said to contain a null value.

• **Primary Key**. can't be null, used to give identity to a row. It must contain a value.

• **Unknown key**. A foreign key is a column or group of columns that references a primary key in the same or another table. The foreign key value must match the corresponding existing primary key value, otherwise it must be null..

- **Foreign Key:-** A Foreign Key is a column or set of columns that refers to a primary key in the same or another table. A foreign key value must match the related existing primary key value or else be null.

4.4 SQL

A special-purpose language designed for data management in relational database management systems (RDBMS). The following types are valid: • NOT NULL • UNIQUE • PRIMARY KEY • FOREIGN KEY • CHECK • DEFAULT

4.4.1 Types of SQL Statement:-

- **Data Definition Language (DDL):-** These statements are used to define the database structure or schema. They are used to manage different objects within the database such as Table, View, sequence, index, constraints etc for e.g. create ,alter etc

- **Data Manipulation Language(DML):-** These statements are required when user wants to add new rows to the table , Update existing rows or Remove rows from the table. That is when user want to manipulate the table.

- **Transaction Control Language(TCL):-** TCL commands are used to handle transactions in the database. These statements come in handy to manage and track the changes being made by the DML queries on our database. For e.g. commit , rollback

- **Data Control Language(DCL):-** It is mainly used for revoke and to grant the user the required access to a database . for e.g. grant , revoke.

4.4.2 Functions:

Functions are very powerful feature of SQL and can be used to perform operations on the data.

Some of the most widely used functions are:-

- Avg()
- Count()
- Max()
- Rank()
- Length()

4.4.3 Operator :-

In order to improve the functionality of condition there by to retrieve user required information we have to use the operator.There are various operator in Sql based of division like Arithematic operator, wildcard operator etc some of them are:-

• **Like operator:-** It is used to determine whether a character column is satisfying specific style or not. The style can be verified by using some special symbols called “wild-card characters” . for e.g. %, _

• **IN/NOT IN operator**:- It is used to verify whether a column is equal or not equal to any one of the list of values specified or not. The values can be specified directly or by using a SELECT statement. For e.g. Select * from emp where sal IN(10000,20000,30000)

• **UNION ALL**: It is used to get combining data between two tables when the both tables are having same structure.

• **INTERSECT**: It is used to get the common data between queries.

4.4.4 Joins :

A join clause is used to combine rows from two or more tables, based on a related column between them. In order to join the tables, they must have a common column. Common column means the names of the columns can differ, but they must have the same data type, size and data.

4.4.4.1 Types of Joins:

• **Inner Join:** This join returns only the matching data between the tables. It is mainly used to retrieve the related data between the tables

• **FULL Outer Join :** retrieves all the rows; matched as well as unmatched from both the tables.

• **Left Outer Join:** Retrieves all rows from the table on the left (LHS), even if there is no match in the table on the right.

• **RIGHT Outer Join:** Retrieves all rows from the table on the right (RHS), even if there is no match in the table on the left.

• **Self Join:** This type of join self-joins the tables. Sometimes users need to join tables to themselves. Example: To find an employee's manager name, the user must join his or her employee table. This is called a self-join.

• **Cross Join:** A join without a common column criterion is called a cross join. This join is used to find different combinations of data between tables. The result is multiplied by the number of records between the tables.

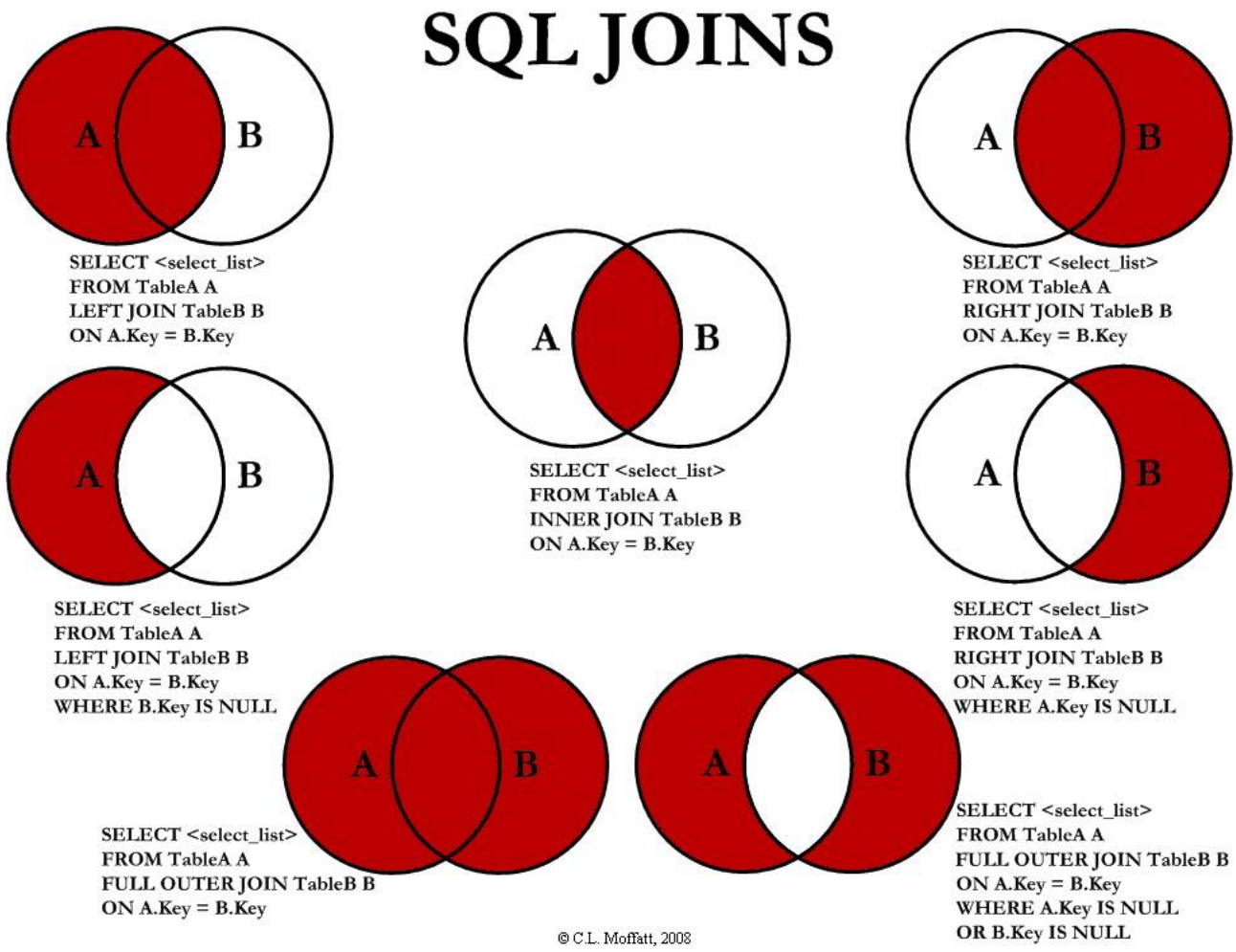


Fig 4.4.4.1 Types of Joins

Chapter 5 : JAVA

5.1 What is java

Java is a programming language and a platform. Java is a high level, robust, objectoriented and secure programming language. Java was developed by Sun Microsystems (which is now the subsidiary of Oracle) in the year 1995. James Gosling is known as the father of Java. Before Java. Platform: Any hardware or software environment in which a program runs, is known as a platform. Since Java has a runtime environment (JRE) and API, it is called a platform.

5.1.1 How is Java Platform Independent ?

The meaning of platformindependent is that the java compiled code(byte code) can run on all operating systems. A program is written in a language that is a humanreadable language. It may contain words, phrases, etc. that the machine does not understand. In order for a machine to understand source code, it must be in a machine-readable language, usually a machine-level language. So comes the role of the compiler.

5.2 Object oriented program (OOPS)

5.2.1 OOPs?

An object-oriented program consists of a group of interacting objects that exchange messages to achieve a common goal. The advantages of OOP are:

• Real programming

• Code reuse

• Code modularity

• Resilience to change

• Information hiding

5.2.2 Encapsulation: Encapsulation is the process of hiding all the details of an object that do not significantly affect its properties.

5.2.3 Abstraction: Abstraction is an intrinsic feature of an object that distinguishes it from all other types of objects and therefore represents a well-defined conceptual boundary related to the viewer's point of view.

5.2.4 Inheritance : Inheritance is taking methods and properties of another class by extending to that class.

5.2.5 Polymorphism • The two Latin words poly mean together and morph means form. • Polymorphism in object-oriented programming refers to a programming language's ability to treat objects differently depending on their data type or class. There are 2 different types of polymorphism present , they are:

• Compile-time polymorphism

• Run-time polymorphism

5.3 Class

• A class is a set p methods and members that are of the same type that of the class.

5.3.1 Objects

• Objects are programming constructs that encapsulate data as program objects with the ability to use or modify data. • An object is a self-contained entity with its own collection of properties (ie data) and methods (ie operations) that encapsulate functionality in a reusable and dynamically loadable structure.

5.3.2 Methods • Actions on objects defined as part of a class declaration. • Methods defined in a class specify what actions the created object can perform.

5.4 Base(Fundamental) Class

5.4.1 Object Classes

• By default, every class we create extends the Object class. That is, the object class is at the top of the hierarchy. This makes it easy to pass objects of any class as arguments to methods.

The finalize() method is called when the object's memory is destroyed.
getClass() it displays the class to which the object belongs.
hashCode() it displays the hash code of the class object.
toString() returns a string corresponding to the object name.
Notification() to send a message to the synced method
wait() pauses the thread for a while.
wait (...) pauses the thread for the specified number of seconds.

5.4.2 Wrapper class

The wrapper class in Java provides the mechanism to convert primitive into object and object into primitive Autoboxing and Unboxing feature convert primitives into objects and objects into primitives automatically. The automatic conversion of primitive into an object is known as autoboxing and vice-versa unboxing. The eight classes of the java.lang package are known as wrapper classes in Java.

5.5 Handling Exceptions

Exception handling is a mechanism for handling run-time errors such as ClassNotFoundException, IOException, SQLException, RemoteException, and more.

5.5.1 Benefits of Exception Handling

The main benefit of exception handling is to maintain normal application flow. Exceptions usually interrupt the normal flow of an application. That's why you need to handle the exception.

CHAPTER 6: MAVEN

6.1 Maven

Maven is a powerful project management tool based on the Project Object Model (POM) used to build projects, dependencies, and documentation. A tool you can use to create and manage Java-based projects. Maven simplifies the daily tasks of Java developers and helps to create and run any Java project.

6.1.1 Maven Lifecycle

Here is the basic Maven lifecycle and its 8 steps: Validate, Compile, Test, Package,

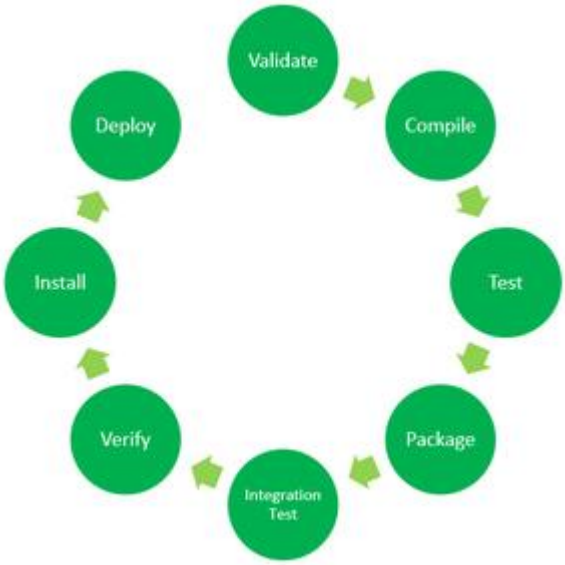


Fig 6.1.1 Maven Lifecycle

The basic Maven lifecycle consists of 8 basic steps or phases for compiling, testing, building, and installing a given Java project:

1. Validate: This step validates that the project structure is correct. For example, make sure all dependencies are downloaded and available in your local repository.

2. Compilation: Compile the source code, convert the .java file to a .class file, and save the class to the target/classes folder.

3. Test: Run unit tests on the project.

4. Package. In this step, the compiled code is packaged into a distributable format such as a JAR or WAR.

5. Integration tests: Run integration tests on your project.

6. Validation: In this step, validation is performed to ensure that the design is valid and meets quality standards.

6.1.2 Maven Commands:

- mvn clean: Cleans the project and removes all files generated by the previous build.
- mvn compile: Compiles source code of the project.
- mvn test-compile: Compiles the test source code.
- mvn test: Runs tests for the project.
- mvn package: Creates JAR or WAR file for the project to convert it into a distributable format.
- mvn install: Deploys the packaged JAR/ WAR file to the local repository.
- mvn deploy: Copies the packaged JAR/ WAR file to the remote repository after compiling, running tests and building the project.

Chapter 7: Spring

7.1 Spring

Spring is one of the most popular enterprise Java frameworks. Developers around the world use Spring to build high quality and reliable applications. It was developed by Rod Johnson in 2003. However, the main disadvantage of the Spring project is that it is time-consuming to install and can be a bit complicated for new developers.

Business(Enterprise) Applications: Software applications created by companies to simplify business processes. Business application deployment requires the deployment of three tiers.

• User interface layer

• Business processing layer

• Data storage and access layer.

User Interface Layer

This layer is the top level of the business application. It provides users with a starting point for interacting with business applications. This provides a very convenient environment for receiving data from users and sending data to server-side applications.

Enterprise Computing Layer: This is the heart of business applications and can be used to define and enforce any business rules and regulations actually required by customers. When developing a business application to prepare for the business processing layer, you must use a separate logic called "business logic". Technologies such as servlets, EJBs, DAOs, etc. must be used to provide the business logic.

Data storage and access level

This is the lowest tier for business applications and provides an excellent environment for interacting with the database to perform archiving operations. Deploying this layer in business applications requires the use of a separate logic called "memory logic". To provide persistence logic you need to use a technology set like JDBC, Hibernate, etc.

7.1.1Features:-

- Loose coupling through dependency injection and interface orientation

- Aspects oriented programming

- Lightweight development with plain old Java objects (POJOs)

- Boilerplate reduction through aspects and templates

7.1.2 Core Module: It is fundamental module in Spring Framework; it has provided basic foundation for all other modules of spring framework. This module can be used to prepare Standalone Applications directly.This module is able to provide the features like IOC Containers, Beans, Dependency Injection.

7.1.3AOP Module[Aspect Oriented Programming]: In general, if we prepare enterprise applications by using only Object Orientation then we have to provide both business logic and Services like Transactions, JMS, JAAS,.... in combined manner, it will provide tightly coupled design, it will provide less sharability and less reusability. In the above context, to improve sharability and Reusability we have to provide loosely coupled design , to get loosely coupled design we have to apply Aspect Oriented Programming.In Aspect Oriented Programming, we will declare each and every service as an aspect and we will inject these aspects in Business Logic at runtime

7.1.4 JDBC/DAO Module: The main purpose of this module is to interact with the database of a Spring application to perform database operations using JDBC Persistance mechanism. The JDBC/DAO module abstracts away from common JDBC implementations, making it easier for Spring applications to interact with the database. By providing a template class, you need to use the following steps to interact with the database in eg JDBC.

1. Driver Download and Registration

2. Establish a connection between the Java application and the database.

3. Write statement/Prepared Statement/Callable Statement according to requirement .

4. Write and run the SQL query.

5. Close Connection In the JDBC steps above, the driver is loaded and registered, the connection is established, the statement is made, and the connection is closed. Common to all JDBC applications, the Spring JDBC/DAO module can provide developers with the ability to abstract frequently used steps and provide mutable parts such as Sql query execution.

7.1.5 ORM Module[Object-Relational Mapping]:

ORM is the mapping between a table, properties or columns and primary key column with the respective bean component provided Bean class name, ID property, normal properties,ORM has define a set of rules and regulations to provide mapping between Object Oriented Data Model and Relational Data Model in order to achieve data persistency.

Eg: Hibernate, JPA,

To prepare Hibernate Application then we have to use the following instructions

1. Create Configuration class object.

2. Create Session Factory class object:

3. Create Session object

4. Perform Persistence operations

5. Close Session Factory and Configuration:

7.1.6 WEB / WEB MVC Module:

The WEB module provided a great environment for integrating other MVC based applications like Struts, JSF. WEB-MVC is an MVC implementation provided directly by Spring for web application deployment. Steps to prepare your first spring application

1. Download Spring Framework from the Internet. 2. Check Spring installation in Eclipse/STS IDE

3. Prepare the empty classroom

4. Prepare the component configuration file

5. Prepare the test/client application.

7.1.7 CIO Container

This is the principle of object-oriented programming, where the objects in the program do not depend on each other.

for a specific implementation of another object

Dependency Injection, also known as DI, is a type of Inversion of Control (IoC).

7.1.8 IOC: Reversal of Control

• The idea of this principle is to redirect the search for resources.

• In a traditional search, the component asks the container to search for the resource, and the container returns accordingly.

concerned resource. • When using IOC, the container itself actively commits resources to the managed bean.

• Members only have to choose how they receive their resources. This can be described as a passive form of navigation.

7.1.9 Dependency Injection

It is a composition of structural design patterns, in which for each function of the application there is one, a conditionally independent object (service) that can have the need to use other objects (dependencies) known to it by interfaces. Dependencies are transferred (implemented) to the service at the time of its creation. This is a situation where we introduce an element of one class into another. In practice, DI is implemented by passing parameters to the constructor or using setters. Libraries that implement this approach are also called IoC containers.

7.1.9.1 Goals of Dependency Injection

- Loose Coupling

- Coding to Interfaces

- Makes the code easy to understand, easy to test and maintain

7.1.9.2 Type of Dependency Injection:

• Setter Injection

• Constructor Injection

• Interface Injection

7.Spring's IOC Container Implementations is of two types

-BeanFactory

-ApplicationContext

7.1.9.3.BeanFactory

--->It is the fundamental or Base container provided by SPring Framework inorder to

manage bean objects.

--->BeanFactory IOC container will provide basic functionalities to the spring framework

by creating maintaining beans objects as per the beans configuration details which we

provided in spring beans configuration file.

Eg 1:

XmlBeanFactory

InputStream is = new FileInputStream("beans.xml");

BeanFactory factory = new XmlBeanFactory(is);

//Get bean

HelloWorld obj =

(HelloWorld)factory.getBean("helloWorld");

Resource resource = new

FileSystemResource("beans.xml");

BeanFactory factory = new XmlBeanFactory(resource);

ClassPathResource resource = new

ClassPathResource("beans.xml");

BeanFactory factory = new XmlBeanFactory(resource);

The Application Context is spring's more advanced container.

• It includes all functionality of the BeanFactory

• Similar to BeanFactory it can load bean definitions, wire beans together and dispense beans upon request.

• Additionally it adds more enterprise-specific functionality.

• Defined by the org.springframework.context.ApplicationContext interface

• BeanFactory is one of its superinterfaces

Eg:2

ApplicationContext

ApplicationContext context = new

FileSystemXmlApplicationContext("beans.xml");

HelloWorld obj = (HelloWorld)

context.getBean("helloWorld");

7.1.10 Implementations of Application Context

• FileSystemXmlApplicationContext

- This container loads the definitions of the beans from an XML file. Here you need to provide the full path of the XML bean

configuration file to the constructor.

• ClassPathXmlApplicationContext

- This container loads the definitions of the beans from an XML file. Here you do not need to provide the full path of the XML file but you need to set CLASSPATH properly because this container will look bean configuration XML file in

7.1.10.1 CLASSPATH.

• XmlWebApplicationContext

- This container loads the XML file with definitions of all beans from within a web application

Constructor or setter injection?

• Constructor Injection: Constructor injection enforces strong dependency contracts. Prevents properties from being changed.

Setter method is not needed, so the number of lines of code is reduced. • Setter Injection If a bean has multiple dependencies, the list of constructor parameters can be quite long. :

When there are multiple ways to create a valid object, it can be difficult to find a unique constructor because constructor signatures differ only in the number and type of parameters. Constructor injection can cause some ambiguity.

7.1.11 Life cycle of components:

Does the IOC container know all the bean definitions in a Spring Framework application?

In the bean configuration file, the IOC container starts the bean.

The next phase of the life cycle.

1. Load the component class

2. Instantiate the component

3. component initialization

4. Destruction of the beans

7.1.11.1. Loading the component class:

In the configuration file, the IOC container contains the specified bean class bytecode.

save to computer

7.1.11.2. Component instantiation

After loading the bean class bytecode into application memory, Spring will IOC

Create an object for an empty class.

7.1.11.3 Initialization and destruction of components:

As part of the bean's lifecycle, the IOC container must initialize the bean. Component instances and IOC containers are responsible for managing component destruction at runtime.

When business logic or IOC containers are disabled

7.1.12 Access to Spring Data

The data access/integration layer consists of JDBC, ORM, OXM, JMS and transaction modules. • The JDBC engine provides a JDBC abstraction layer, eliminating the need for lengthy JDBC code and parsing.

The error code for the specific database provider. • ORM modules: JPA, JDO, Hibernate,

. The ORM package allows you to use all of these O/R mapping configurations in conjunction with everything else.

Features provided by Spring, such as the simple declarative transaction management feature mentioned earlier. • OXM Module – JAXB.

XMLBeans and XStream. • The Java Messaging Service (JMS) module contains functions for creating and using messages. • The transaction mechanism supports programmatic and declarative transaction management for classes that implement:

For special interfaces and all common Java objects (POJOs). Connection pool data source

```
<bean id="dataSource“
class="org.apache.commons.dbcp.BasicDataSource"
>
<property name="driverClassName"
value="org.hsqldb.jdbcDriver" />
<property name="url“
value="jdbc:hsqldb:hsql://localhost/roadrantz/roadrantz"
/>
<property name="username" value="sa" />
<property name="password" value="" />
<property name="initialSize" value="5" />
<property name="maxActive" value="10" />
</bean>
```

7.1.13 ORM

• The ORM engine provides an integration layer for popular object-relational mapping APIs, including JPA, JDO, Hibernate and iBatis.

•JPA

The Java Persistence API provides specifications for persisting, reading, and manipulating data in Java objects such as relational tables in a database.

Hibernate is an object-relational mapping solution for the Java environment. Hibernate is a Java-based ORM tool that provides a framework for mapping application domain objects to relational

database tables and vice versa. Hibernate provides a reference implementation of the Java Persistence API, making it an excellent choice as an ORM tool that benefits from free coupling.

7.2 Spring Boot

• Spring 5 supports Spring Boot 2, but first let's try to understand what Spring Boot is.

• Spring Boot is a module of the Spring Framework that provides RAD (Rapid Application Development) functions for the Spring Framework. • It relies heavily on the functionality of the original model, which is very powerful and fully functional.

7.2.1 Spring Boot - Spring Boot Starter

• Spring Boot Launcher is a template containing a set of all relevant transitive dependencies required for execution.

• For example, if you want to build a Spring WebMVC application, the existing installation will take care of that

The addictions you need yourself.

• This leads to version conflicts, which eventually lead to several runtime exceptions. • If you start Spring to build an MVC application, all you have to do is import the spring-boot-starter-web dependency.

7.2.1.1 Autoconfiguration is enabled with @EnableAutoConfiguration annotation.

• Spring boot auto configuration scans the classpath, finds the libraries in the classpath and then attempt to guess the best

configuration for them, and finally configure all such beans.

• Auto-configuration tries to be as intelligent as possible and will back-away as you define more of your own configuration.

• Spring boot applications always include tomcat as embedded server dependency.

• You can exclude tomcat and include any other embedded server if you want. Or you can make exclude server environment altogether. It’s all configuration based.

7.2.2 Spring Boot - Bootstrap the application

• To run the application, we need to use @SpringBootApplication annotation.

• Behind the scenes, that's equivalent to @Configuration, @EnableAutoConfiguration, and @ComponentScan together.

• It enables the scanning of config classes, files and load them into spring context

7.2.2.1 Benefits of Spring Boot

• Spring Boot helps you resolve addiction conflicts. Determine the required dependencies and import them. • Contains compatible version information for all dependencies. This minimizes problems with classloaders at runtime. • This "basic setup approved" approach helps you organize what matters most behind the scenes. to assert oneself

You can only use it when you need it. everything else works fine. This is the standard code, comments and

XML configuration. • Provides an integrated Tomcat HTTP server for rapid development and testing.

7.2.3Spring AOP

• A framework based on the interfaces of the op Alliance. • Aspects are coded in pure Java code. There is no need to learn the Pointcut query language available in other AOPs.

• Aspects of Spring is customizable with your IoC container. - Objects received from the IoC container can be transparently recommended based on the settings.

• Spring AOP has built-in features like transaction management, performance monitoring, etc.

Bean

• Spring AOP is not as powerful as other implementations like AspectJ.

- However, it supports using common aspects to solve common problems in business applications.

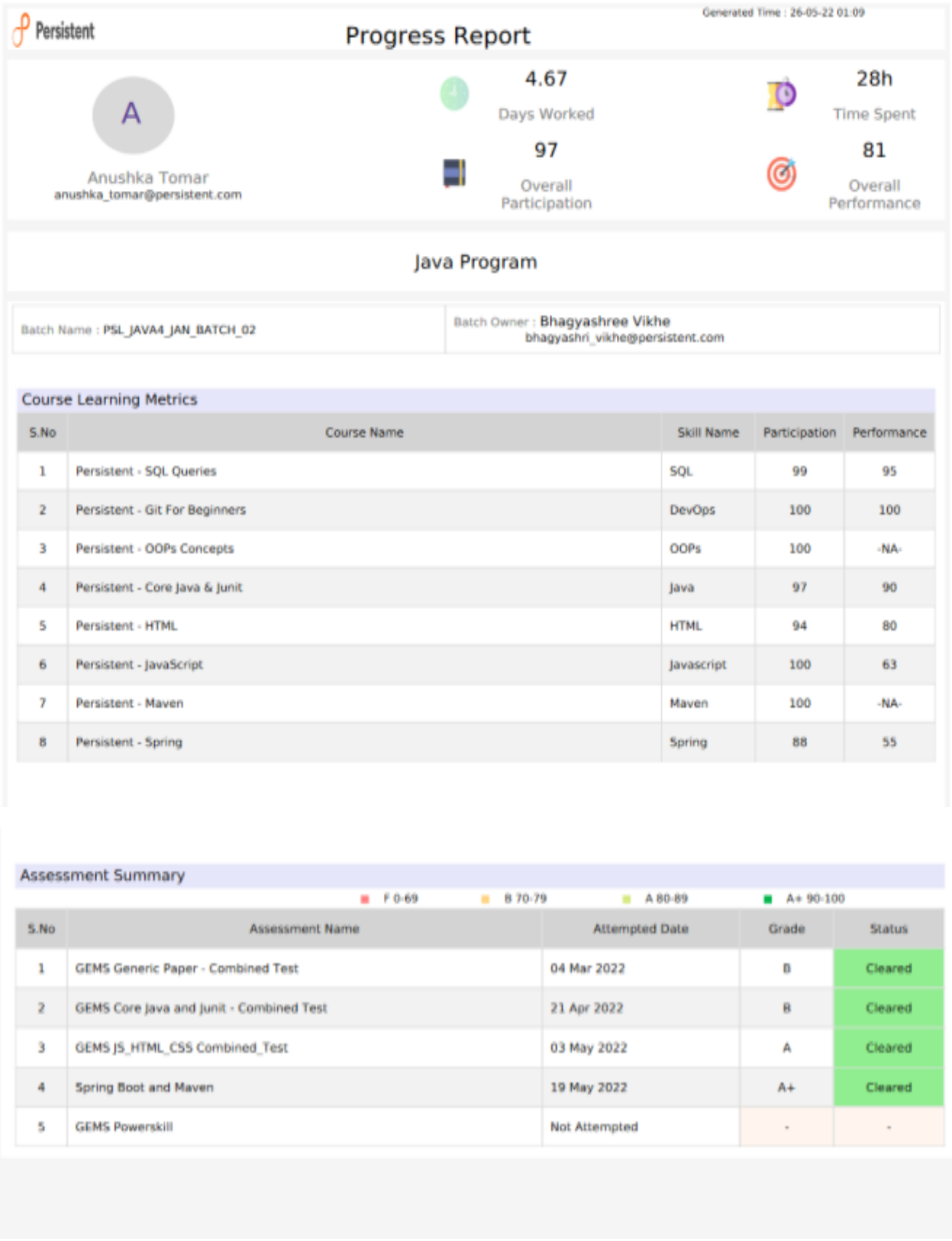
Chapter 8: Conclusion

The internship enabled us to execute a desired output with the help of the various technologies which were taught in the overall process. Apart from these various important technologies, the internship also taught us the value of team work through various events and the importance of ethics in the corporate world through some live lectures. Persistent believes that the pillar of success include four major stones Confident, Responsible, Persistent and Ingenious. The internship helped us in inculcating various abilities so as to stand tall in the domain assigned to us in future and the corporate world, in general..

During my Internship, I have completed all the mentioned modules in the above chapters. The assessments of the modules are still going on and I have cleared all of the assessments till now.

There allocation of Projects will get initiated soon and candidates will be assigned to them according to the requirement of business units of the Organization.

Progress Report



Persistent

Generated Time : 26-05-22 01:09

Progress Report

A

Anushka Tomar
anushka_tomar@persistent.com

4.67
Days Worked

28h
Time Spent

97
Overall Participation

81
Overall Performance

Java Program

Batch Name : PSL_JAVA4_JAN_BATCH_02

Batch Owner : Bhagyashree Vikhe
bhagyashri_vikhe@persistent.com

Course Learning Metrics

S.No	Course Name	Skill Name	Participation	Performance
1	Persistent - SQL Queries	SQL	99	95
2	Persistent - Git For Beginners	DevOps	100	100
3	Persistent - OOPs Concepts	OOPs	100	-NA-
4	Persistent - Core Java & Junit	Java	97	90
5	Persistent - HTML	HTML	94	80
6	Persistent - JavaScript	Javascript	100	63
7	Persistent - Maven	Maven	100	-NA-
8	Persistent - Spring	Spring	88	55

Assessment Summary

F 0-69 B 70-79 A 80-89 A+ 90-100

S.No	Assessment Name	Attempted Date	Grade	Status
1	GEMS Generic Paper - Combined Test	04 Mar 2022	B	Cleared
2	GEMS Core Java and Junit - Combined Test	21 Apr 2022	B	Cleared
3	GEMS JS_HTML_CSS Combined_Test	03 May 2022	A	Cleared
4	Spring Boot and Maven	19 May 2022	A+	Cleared
5	GEMS Powerskill	Not Attempted	-	-

References:

- E-box platform
- Udemy courses
- Apna College(https://youtu.be/bSrm9RXwBaI)
- Persistent Material

Appendices

FPR:

All FPRs are attested from next page.

Name: Anushka Singh Tomar

Enrollment No.: 0901CS181016

Company: Persistent Systems Ltd

FPR-1 from 12th January 2022 to 13th February 2022

Week 1 : 12 Jan - 16 Jan

- Induction program
- About the company quiz
- Soft skills quiz
- Technical quiz

Week 2: 17 Jan- 23 Jan

- Git training
- Assignment to create a git program
- Git self learning quiz

Week 3: 24 Jan- 30 Jan

- OOPS training
- Assignments to create a functional diagram of a program
- OOPS quiz

Week 4: 31 Jan – 6 Feb

- Software testing basics
- Individual assignment to create an excel sheet of testing:
    - Search function on any musical application
    - Add to cart functionality of e commerce website
    - Money Transfer functon on any netbanking application
    - Quality attributes for software testing
- Group assignment to create an excel sheet to test the application ‘Wordpress’
- Software testing quiz.

Week 5: 7 Feb- 13 Feb

- SQL self-learning videos
- SQL quiz

FORTNIGHTLY PROGRESS REPORT (FPR)-2 FROM INDUSTRY MENTOR

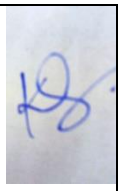
Name of student	Anushka Singh Tomar		Department	CSE	
Industry/Organization	Persistent Systems		Date/Duration	13/02/22-02/03/22	
Criterion	**Poor**	**Average**	**Good**	**Very Good**	**Excellent**
Punctuality/Timely completion of assigned work			✓		
Learning capacity/Knowledge up gradation				✓	
Performance/Quality of work				✓	
Behaviour/Discipline/Team work				✓	
Sincerity/Hard work				✓	
Comment on nature of work done/Area/Topic	Core Java, JDBK, Junit				
OVERALL GRADE (Any one)	**VERY GOOD**				
Name of Industry Mentor	Jayati Munot				
Signature of Industry Mentor	Jayati Munot				

Receiving Date		**Name of Faculty Mentor**	Prof. Khushboo Agarwal	**Sign**	

FORTNIGHTLY PROGRESS REPORT (FPR)-3 FROM INDUSTRY MENTOR

Name of student	Anushka Singh Tomar		Department	CSE	
Industry/Organization	Persistent Systems		Date/Duration	03/03/22-15/03/22	
Criterion	**Poor**	**Average**	**Good**	**Very Good**	**Excellent**
Punctuality/Timely completion of assigned work				✓	
Learning capacity/Knowledge up gradation				✓	
Performance/Quality of work					✓
Behaviour/Discipline/Team work				✓	
Sincerity/Hard work				✓	
Comment on nature of work done/Area/Topic	JDBC programming, Mockito, Maven				
OVERALL GRADE (Any one)	**VERY GOOD**				
Name of Industry Mentor	Jayati Munot				

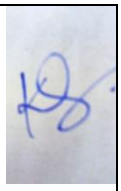
Signature of Industry Mentor	Jayati Munot

Receiving Date		**Name of Faculty Mentor**	Prof. Khushboo Agarwal	**Sign**	

FORTNIGHTLY PROGRESS REPORT (FPR)-4 FROM INDUSTRY MENTOR

Name of student	Anushka Singh Tomar		Department	CSE	
Industry/Organization	Persistent Systems Ltd.		Date/Duration	16/03/22-30/03/22	
Criterion	**Poor**	**Average**	**Good**	**Very Good**	**Excellent**
Punctuality/Timely completion of assigned work					✓
Learning capacity/Knowledge up gradation			✓		
Performance/Quality of work			✓		
Behaviour/Discipline/Team work			✓		
Sincerity/Hard work				✓	

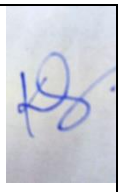
Comment on nature of work done/Area/Topic	Maven, Spring, HTML, CSS
OVERALL GRADE (Any one)	**VERY GOOD**
Name of Industry Mentor	Jayati Munot
Signature of Industry Mentor	Jayati Munot

Receiving Date		**Name of Faculty Mentor**	Prof. Khushboo Agarwal	**Sign**	

FORTNIGHTLY PROGRESS REPORT (FPR)-5 FROM INDUSTRY MENTOR

Name of student	Anushka Singh Tomar		Department	CSE	
Industry/Organization	Persistent Systems Ltd		Date/Duration	31/03/22-15/04/22	
Criterion	**Poor**	**Average**	**Good**	**Very Good**	**Excellent**
Punctuality/Timely completion of assigned work				✓	
Learning capacity/Knowledge up gradation			✓		
Performance/Quality of work					✓

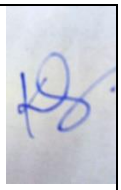
Behaviour/Discipline/Team work				✓	
Sincerity/Hard work					✓
Comment on nature of work done/Area/Topic	HTML, CSS, JAVASCRIPT, BOOTSTRAP				
OVERALL GRADE (Any one)	**VERY GOOD**				
Name of Industry Mentor	Jayati Munot				
Signature of Industry Mentor	Jayati Munot				

Receiving Date		**Name of Faculty Mentor**	Prof. Khushboo Agarwal	**Sign**	

FORTNIGHTLY PROGRESS REPORT (FPR)-6 FROM INDUSTRY MENTOR

Name of student	Anushka Singh Tomar		Department	CSE	
Industry/Organization	Persistent Systems		Date/Duration	16/04/22-31/04/22	
Criterion	**Poor**	**Average**	**Good**	**Very Good**	**Excellent**
Punctuality/Timely completion of assigned			✓		

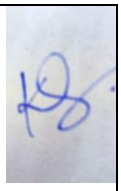
work					
Learning capacity/Knowledge up gradation				✓	
Performance/Quality of work					✓
Behaviour/Discipline/Team work				✓	
Sincerity/Hard work				✓	
Comment on nature of work done/Area/Topic	Advanced SQL				
OVERALL GRADE (Any one)	**VERY GOOD**				
Name of Industry Mentor	Jayati Munot				
Signature of Industry Mentor	Jayati Munot				

Receiving Date		**Name of Faculty Mentor**	Prof. Khushboo Agarwal	**Sign**	

FORTNIGHTLY PROGRESS REPORT (FPR)-7 FROM INDUSTRY MENTOR

Name of student	Anushka Singh Tomar	Department	CSE

Industry/Organization	Persistent Systems		Date/Duration	16/04/22-31/04/22	
Criterion	**Poor**	**Average**	**Good**	**Very Good**	**Excellent**
Punctuality/Timely completion of assigned work			✓		
Learning capacity/Knowledge up gradation			✓		
Performance/Quality of work					✓
Behaviour/Discipline/Team work				✓	
Sincerity/Hard work				✓	
Comment on nature of work done/Area/Topic	Objective Assessments				
OVERALL GRADE (Any one)	**VERY GOOD**				
Name of Industry Mentor	Jayati Munot				
Signature of Industry Mentor	Jayati Munot				

Receiving Date		**Name of Faculty Mentor**	Prof. Khushboo Agarwal	**Sign**	

FORTNIGHTLY PROGRESS REPORT (FPR)-8 FROM INDUSTRY MENTOR

Name of student	Anushka Singh Tomar		Department	CSE	
Industry/Organization	Persistent Systems		Date/Duration	1/05/22-15/05/22	
Criterion	**Poor**	**Average**	**Good**	**Very Good**	**Excellent**
Punctuality/Timely completion of assigned work			✓		
Learning capacity/Knowledge up gradation				✓	
Performance/Quality of work					✓
Behaviour/Discipline/Team work				✓	
Sincerity/Hard work				✓	
Comment on nature of work done/Area/Topic	Objective Assessments				
OVERALL GRADE (Any one)	**VERY GOOD**				
Name of Industry Mentor	Jayati Munot				
Signature of Industry Mentor	Jayati Munot				

Receiving Date		**Name of Faculty Mentor**	Prof. Khushboo Agarwal	**Sign**