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:

ANUBHAV GUPTA

0901CS181014

Faculty Mentor:

Prof. Khushboo Agarwal, Assistant Professor,

Computer Science and Engineering

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 fulfillment of the requirement for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

Submitted by:

ANUBHAV GUPTA

0901CS181014

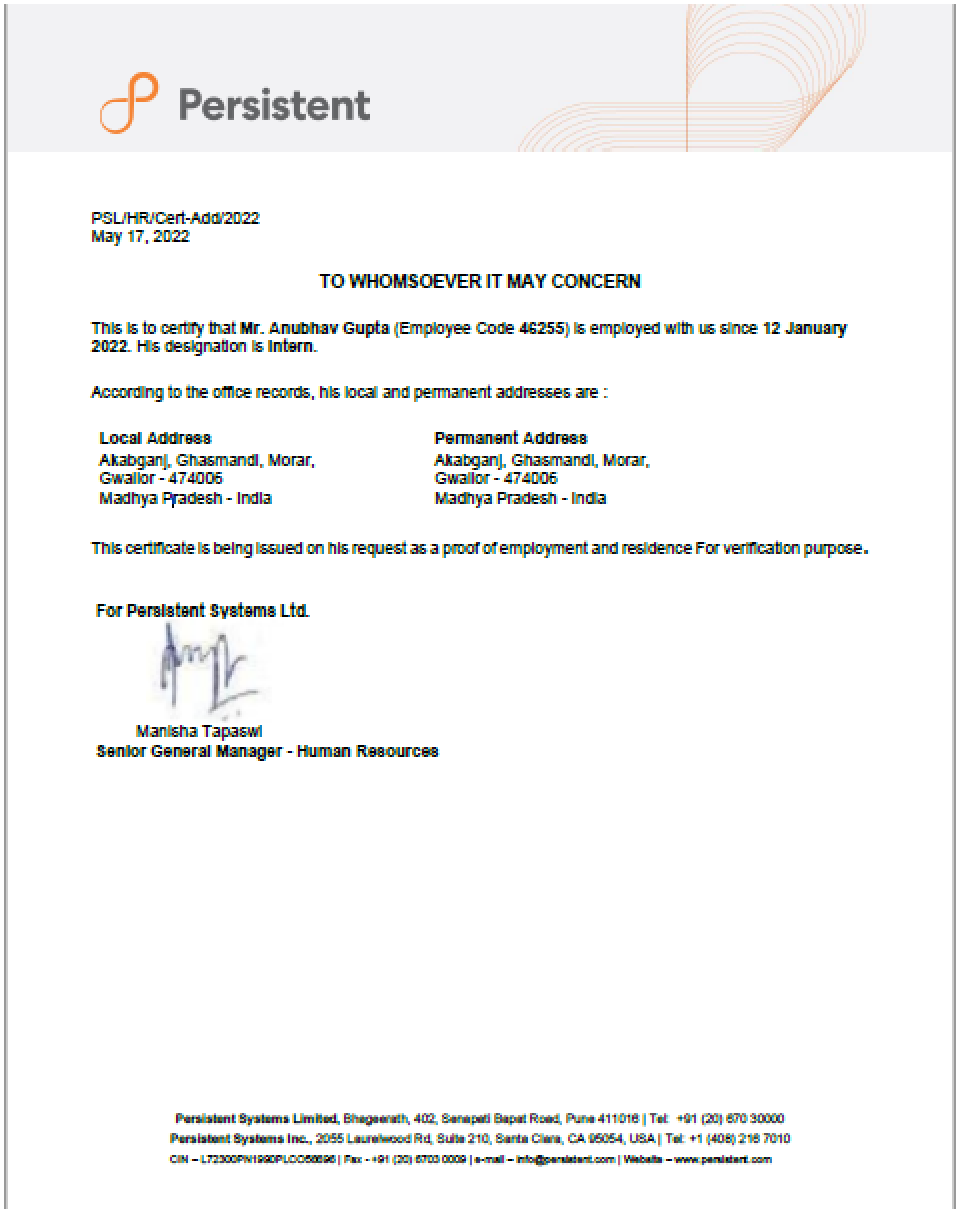
Internship Faculty Mentor:

Prof. Khushboo Agarwal, Assistant Professor,
Computer Science and Engineering

Submitted to:

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE
GWALIOR - 474005 (MP) est. 1957

MAY-JUNE 2022



Persistent

PSL/HR/Cert-Add/2022
May 17, 2022

TO WHOMSOEVER IT MAY CONCERN

This is to certify that **Mr. Anubhav Gupta** (Employee Code **46255**) is employed with us since **12 January 2022.** His designation is **Intern.**

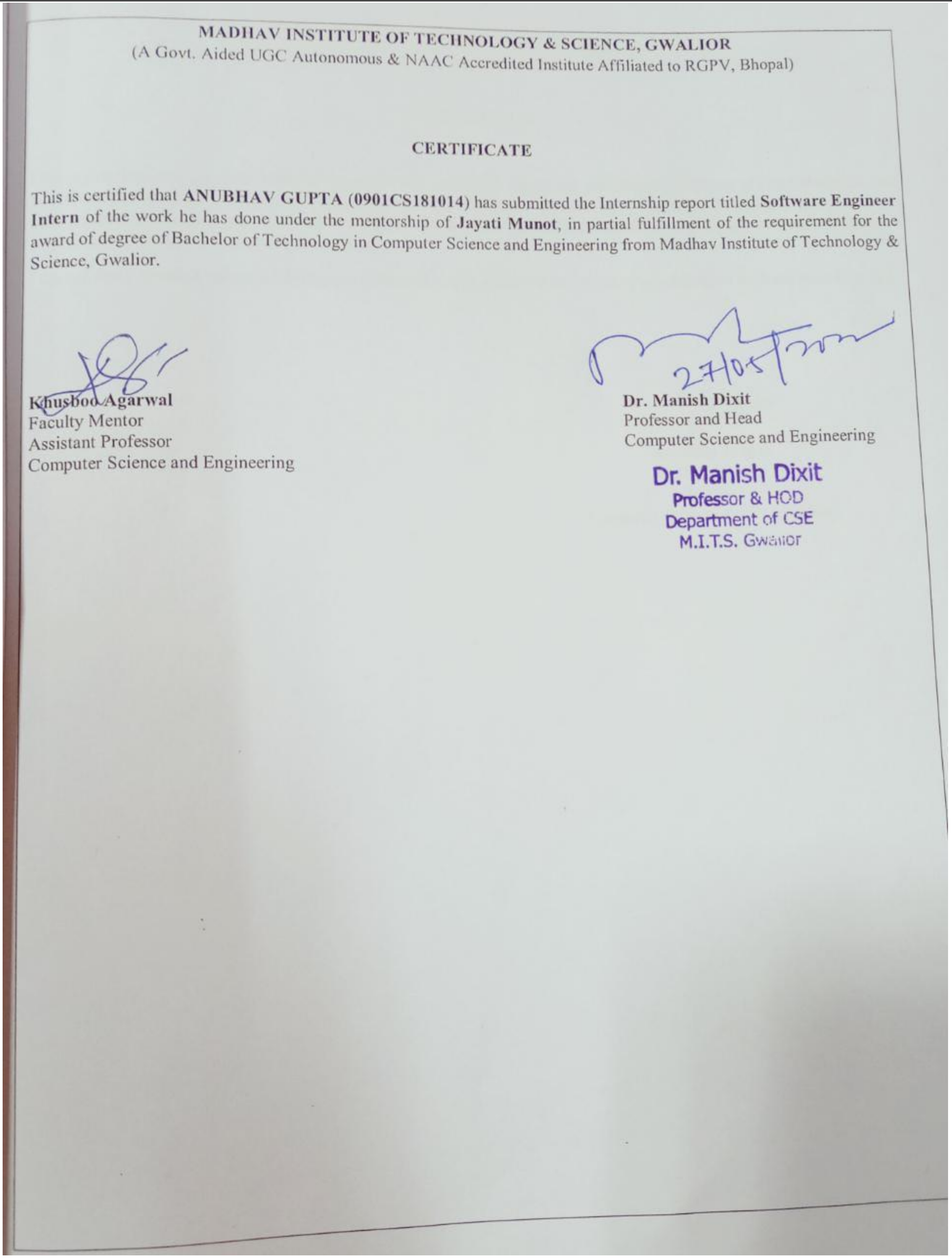
According to the office records, his local and permanent addresses are :

Local Address	**Permanent Address**
Akabganj, Ghasmandi, Morar, Gwalior - 474006 Madhya Pradesh - India	Akabganj, Ghasmandi, Morar, Gwalior - 474006 Madhya Pradesh - India

This certificate is being issued on his 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 **ANUBHAV GUPTA (0901CS181014)** has submitted the Internship report titled **Software Engineer Intern** of the work he has done under the mentorship of **Jayati Munot**, in partial fulfillment of the requirement for the award of degree of Bachelor of Technology in Computer Science and Engineering from Madhav Institute of Technology & Science, Gwalior.

Khusboo Agarwal
Faculty Mentor
Assistant Professor
Computer Science and Engineering

27/05

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 fulfillment 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. Khushboo Agarwal, Assistant Professor**, Department of Computer Science & Engineering.

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

Anubhav Gupta
0901CS181014
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 & 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, Assistant 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.

Anubhav Gupta
0901CS181014
IV Year,
Computer Science and Engineering

ABSTRACT

An Industrial training is an vital segment of a scholar life. A nicely planned, well carried out and evaluated commercial schooling facilitates plenty in growing a expert attitude. It develops an attention of commercial method to problem-solving, primarily based totally on a wide know-how of procedure and mode of operation of organization. The intention and motivation of this commercial schooling is to obtain discipline, abilities, teamwork and technical expertise via a right schooling environment, with a purpose to assist me, as a scholar with inside the area of Information Technology, to broaden a responsiveness of the self-disciplinary nature of troubles in statistics and verbal exchange technology. During a length of six months schooling at Persistent Systems, I changed into assigned to examine numerous publications and abilities and technology as in line with required with the aid of using the enterprise to paintings easily and efficiently. As a result, I crucial to reap the necessities to paintings on this following enterprise with healthy of expertise and enjoy with inside the required technology. Throughout this commercial schooling, I were found out new programming language that required for the system, the procedure of the manufacturing strains and capable of put in force what I actually have found out for the beyond yr as a scholar. My internship is essentially a mixture of phases, Phase 1 consisted of Learning & Development Department whilst Phase 2 includes Live projects. Currently, we're present process in Phase 1 which accommodates mastering exclusive modules which shape a key element to construct any software.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1: INTRODUCTION

As an Academic and Software Engineer Intern, we had 2 phases. Phase 1 consisted of Learning & Development Department while Phase 2 consists of Live projects. Currently, we are undergoing in Phase 1 which consists of 8 learning modules which form a key component to build any software. Git, SQL, OOPS, Java, Spring, Maven, HTML, CSS and JavaScript. After that, we will be allocated to advance training modules with Live Projects.

Chapter 2: Technologies Studied

2.1 Back-End Technologies

2.1.1 SQL

SQL is a database computer language designed for the retrieval and management of data in a relational database. SQL stands for Structured Query Language.

RDBMS

RDBMS stands for Relational Database Management System.

RDBMS is the basis for SQL, and for all modern database systems such as MS SQL Server, IBM DB2, Oracle, MySQL, and Microsoft Access.

The data in RDBMS is stored in database objects called tables. A table is a collection of related data entries and it consists of columns and rows.

Applications of SQL **:**

- Allows users to access data in the relational database management systems.
- Allows users to describe the data.
- Allows users to define the data in a database and manipulate that data.
- Allows to embed within other languages using SQL modules, libraries & pre-compilers.
- Allows users to create and drop databases and tables.
- Allows users to create view, stored procedure, functions in a database.
- Allows users to set permissions on tables, procedures and views.

SQL Joins:

- (INNER) JOIN: Returns records that have matching values in both tables
- LEFT (OUTER) JOIN: Returns all records from the left table, and the matched records from the right table
- RIGHT (OUTER) JOIN: Returns all records from the right table, and the matched records from the left table
- FULL (OUTER) JOIN: Returns all records when there is a match in either left or right table

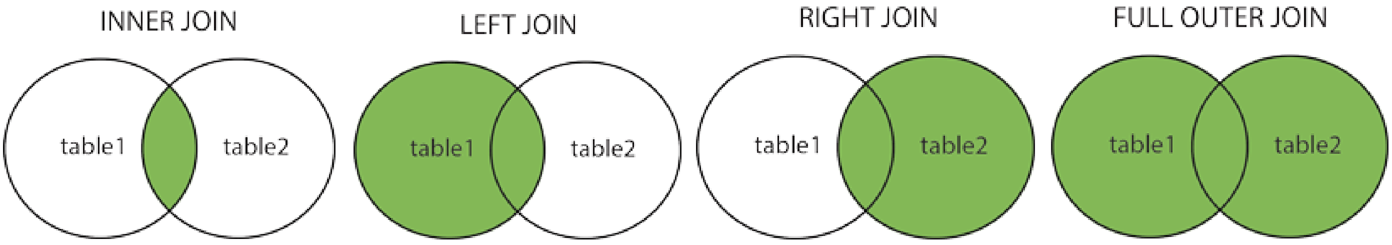


Fig.1 Types of Joins in DBMS

Types of Database Languages:

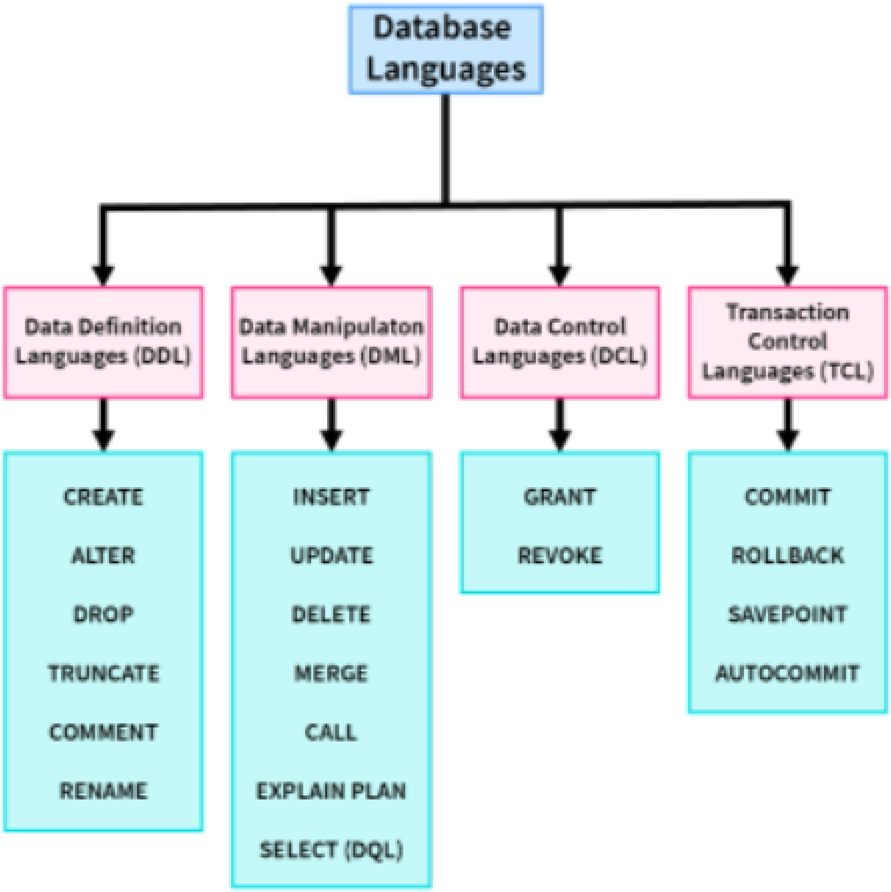


Fig. 2 Types of Database Languages

1. Data definition language (DDL):

- CREATE: Creates a new database or object, such as a table, index or column
- ALTER: Changes the structure of the database or object
- DROP: Deletes the database or existing objects
- RENAME: Renames the database or existing objects
- TRUNCATE: It is used to remove all records from a table.
- RENAME: It is used to rename an object.

2. Data manipulation language (DML):

- INSERT: Adds new data to the existing database table
- UPDATE: Changes or updates values in the table
- DELETE: Removes records or rows from the table
- SELECT: Retrieves data from the table or multiple tables
- MERGE: It performs UPSERT operation, i.e., insert or update operations.
- CALL: It is used to call a structured query language or a Java subprogram.
- EXPLAIN PLAN: It has the parameter of explaining data.

3. Data control language (DCL):

- GRANT: Gives a user access to the database

- REVOKE: Removes a user's access to the database

4. Transaction control language (TCL):

- COMMIT: Carries out a transaction
- ROLLBACK: Restores a transaction if any tasks fail to execute
- SAVEPOINT: Sets a point in a transaction to save.
- AUTOCOMMIT: It is used to enable/disable the auto-commit process that commits each transaction after its execution.

TYPES OF RELATIONSHIPS IN DBMS:

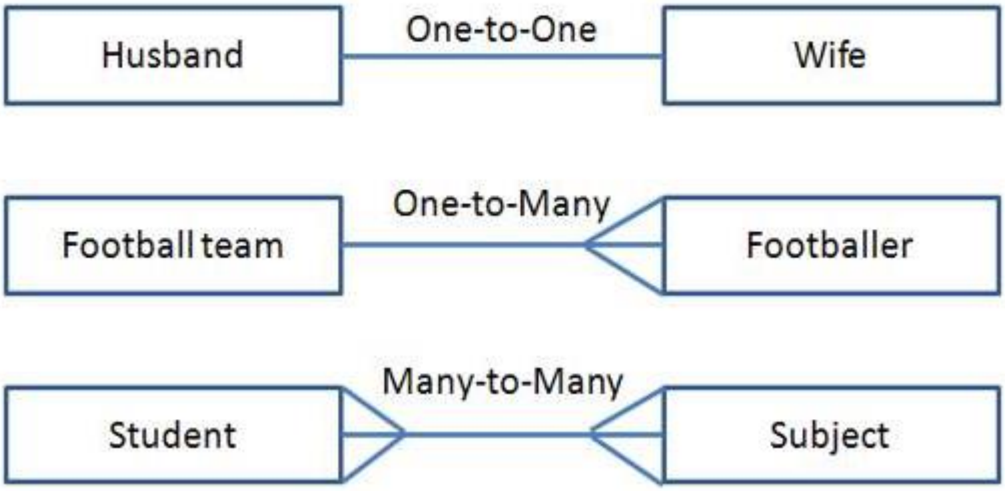


Fig.3 Types of Relationship in DBMS

CONSTRAINTS IN DBMS:

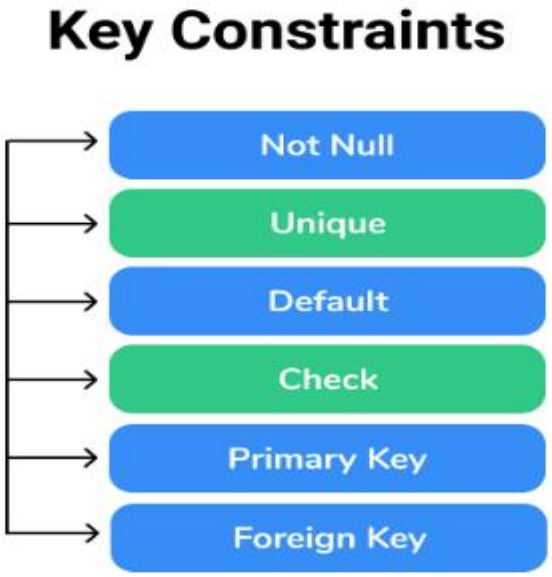


Fig.4 Types of Constraints in DBMS

2.1.2 OOPS:

OBJECT ORIENTED PROGRAMMING SYSTEM

Object-Oriented Programming or OOPs refers to languages that uses objects in programming. Object-oriented programming aims to implement real-world entities like inheritance, hiding, polymorphism etc in programming. The main aim of OOP is to bind together the data and the functions that operate on them so that no other part of the code can access this data except that function.

OOPS CONCEPT:

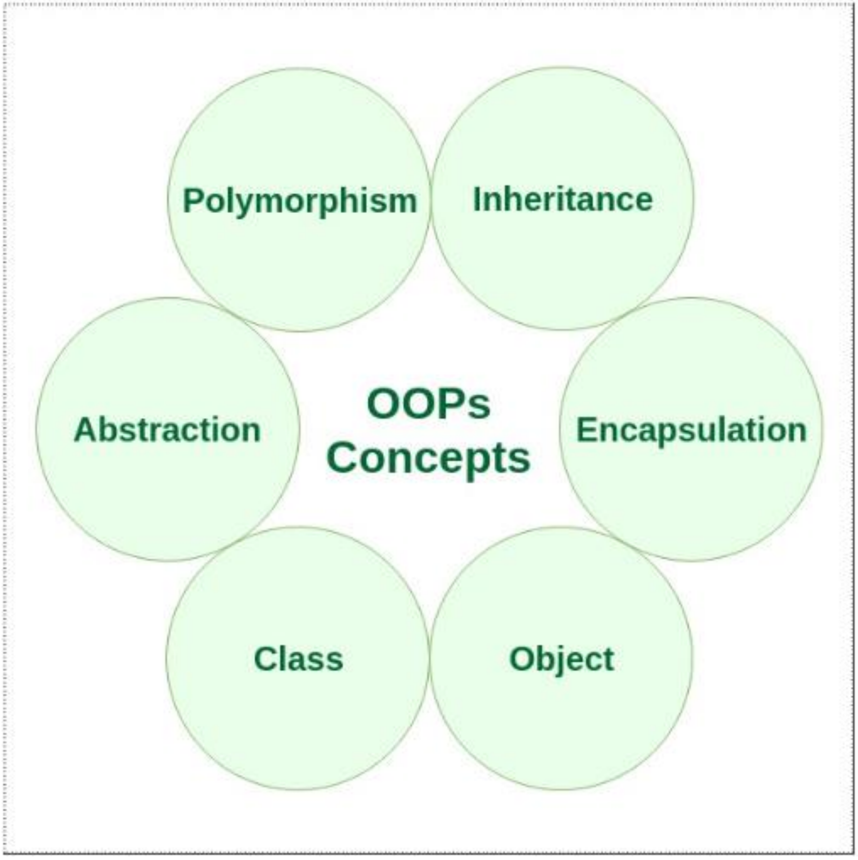


Fig. 5 OOPS Concept

Apart from these concepts, there are some other terms which are used in Object-Oriented design:

- Coupling
- Cohesion
- Association
- Aggregation
- Composition

1) Class

The class is one of the Basic concepts of OOPs which is a group of similar entities. It is only a logical component and not the physical entity.

Example: if you had a class called "Expensive Cars" it could have objects like Mercedes, BMW, Toyota, etc. Its properties(data) can be price or speed of these cars. While the methods may be performed with these cars are driving, reverse, braking etc.

2) Object

An object can be defined as an instance of a class, and there can be multiple instances of a class in a program.

For example – chair, bike, marker, pen, table, car, etc.

3) Inheritance

Inheritance is one of the Basic Concepts of OOPs in which one object acquires the properties and behaviors of the parent object. It's creating a parent-child relationship between two classes.

4) Polymorphism

Polymorphism refers to one of the OOPs concepts in Java which is the ability of a variable, object or function to take on multiple forms.

5) Abstraction

Abstraction is one of the OOP Concepts which is an act of representing essential features without including background details.

Example: while driving a car, you do not have to be concerned with its internal working. Here you just need to concern about parts like steering wheel, Gears, accelerator, etc.

6) Encapsulation

Encapsulation is one of the best concepts of wrapping the data and code. In this OOPs concept, the variables of a class are always hidden from other classes.

7) Association

Association is a relationship between two objects. It is one of the OOP Concepts which defines the diversity between objects. In this OOP concept, all objects have their separate lifecycle, and there is no owner. For example, many students can associate with one teacher while one student can also associate with multiple teachers.

8) Aggregation

In this technique, all objects have their separate lifecycle. However, there is ownership such that child object can't belong to another parent object. For example consider class/objects department and teacher. Here, a single teacher can't belong to multiple departments, but even if we delete the department, the teacher object will never be destroyed.

9) Composition

Composition is a specialized form of Aggregation. It is also called "death" relationship. Child objects do not have their life-cycle so when parent object deletes all child object will also delete automatically. For that, let's take an example of

House and rooms. Any house can have several rooms. One room can’t become part of two different houses. So, if you delete the house room will also be deleted.

Advantages of OOPs (Object-Oriented Programming System):

- OOPs Concepts offer easy to understand and a clear modular structure for programs.
- Objects created for Object-Oriented Programs can be reused in other programs. Thus it saves significant development cost.
- Large programs are difficult to write, but if the development and designing team follow OOPS concepts, then they can better design with minimum flaws.
- It enhances program modularity because every object exists independently.

2.1.3 GIT AND GITHUB

2.1 Introduction to GIT

Git is a software for tracking changes in any set of files, usually used for coordinating work among programmers collaboratively developing source code during software development. Its goals include speed, data integrity, and support for distributed, non-linear workflows (thousands of parallel branches running on different systems).

It outclasses SCM equipment(tools) like Subversion, CVS, Perforce, and Clear Case with functions like multiple workflows , convenient staging areas and cheap local branching.

Git comes with Integrated GUI tools (git-gui, gitk), however there are several third-party tools for users seeking for a platform--specific experience.

Advantages:

- Fast Processing: As compared to other software or apps, it runs more quickly. Both the server as well as local operations are performed easily with high speed.
- Flexible: The work-flow operations in the system are flexible in nature. It is possible to make a choice from the work-flow options.
- Easy Merging: It is possible to start merging another code in the system. It is a great way for the developers to interact with each other and add to their contributions. No long procedure is required to follow during the merging time.

Disadvantages:

Not Suitable for Binary Files:- It fails with the presence of files having binary data. It starts processing every work slowly. Any file which doesn’t support textual data is not compatible with this technology.

No Sub-Trees Checkout Possible:- In this system, it is not supported to check the sub-trees. For checking each particular project, the need to create multiple repositories with pre-package arise.

2.2 Version Control System (VCS):

Version Control Systems are the software tools for tracking/managing all the changes made to the source code during the project development. It keeps a record of every single change made to the code. It also allows us to turn back to the previous version of the code if any mistake is made in the current version. Without a VCS in place, it would not be possible to monitor the development of the project.

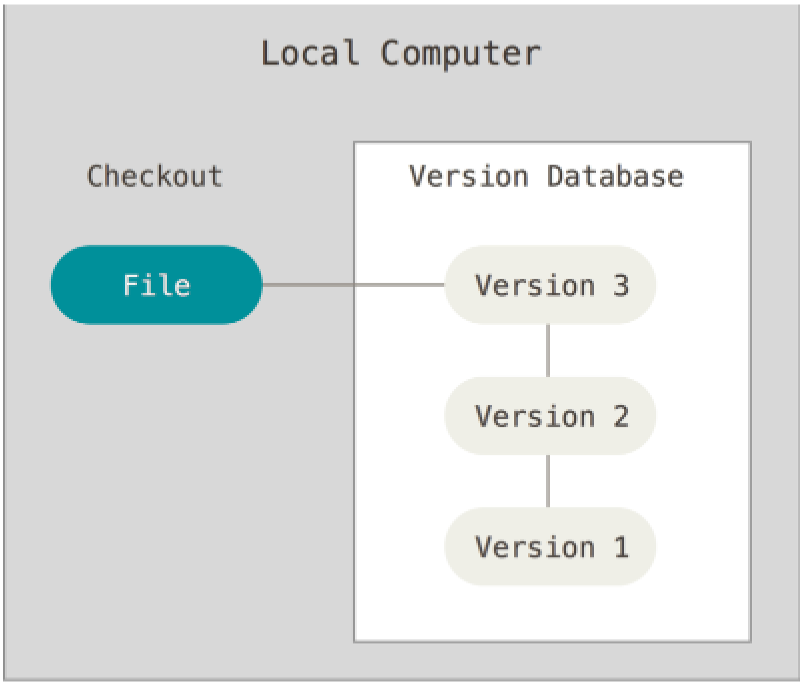
Types of VCS

The three types of VCS are:

1. Local Version Control System
2. Centralized Version Control System
3. Distributed Version Control System

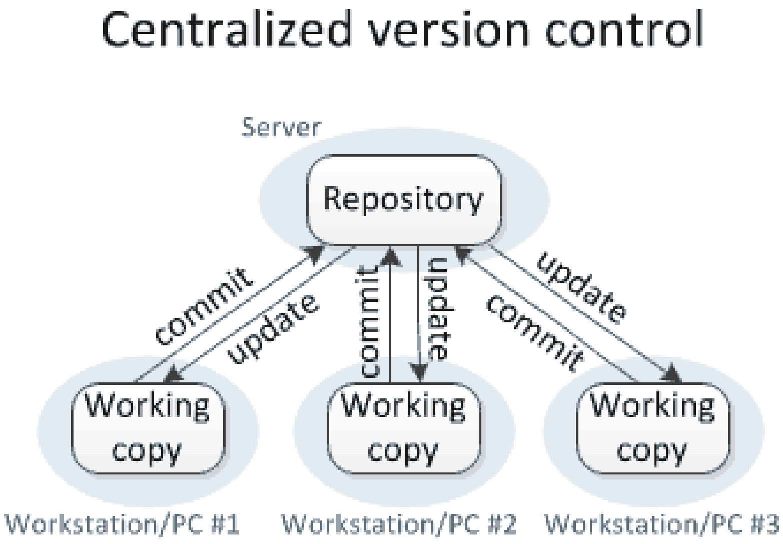
Local Version Control System:

Local Version Control System is located in your local machine. If the local machine crashes, it would not be possible to retrieve the files, and all the information will be lost. If anything happens to a single version, all the versions made after that will be lost. Also, with the Local Version Control System, it is not possible to collaborate with other collaborators.To collaborate with other developers on other systems, Centralized Version Control Systems are developed.



Centralized Version Control System:

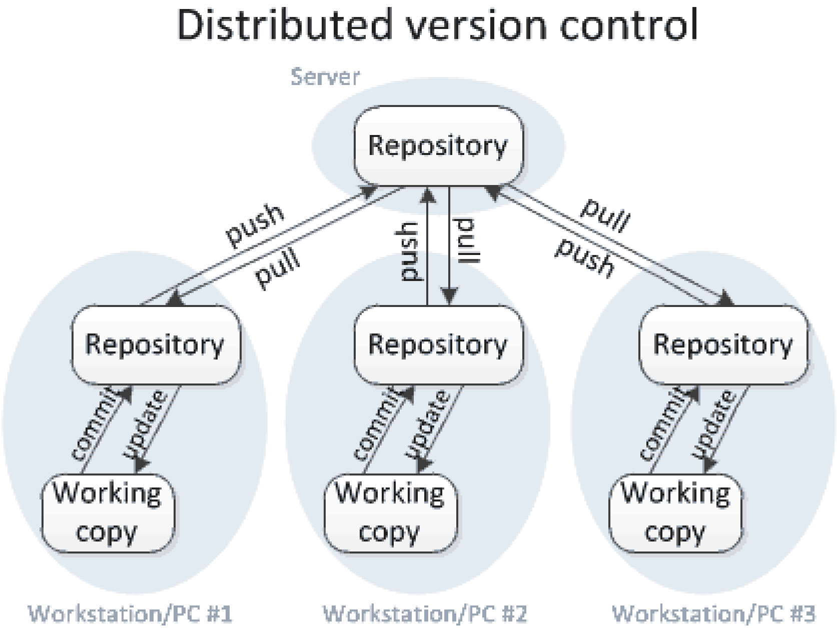
In the Centralized Version Control Systems, there will be a single central server that contains all the files related to the project, and many collaborators checkout files from this single server (you will only have a working copy). The problem with the Centralized Version Control Systems is if the central server crashes, almost everything related to the project will be lost.To overcome all the above problems, Distributed Version Control Systems are developed.



Distributed Version Control System:

In a distributed version control system, there will be one or more servers and many collaborators similar to the centralized system. But the difference is, not only do they check out the latest version, but each collaborator will have an exact copy (mirroring) of the main repository(including its entire history) on their local machines.

Each user has their own repository and a working copy. This is very useful because even if the server crashes we would not lose everything as several copies are residing in several other computers.



2.2 Version Control System(VCS) :

Version control system also called as source control, it is the practice of managing changes and tracking of the software code. Vcs are software tools manage changes to source code over time which help software teams. As development environments have accelerated, vcs help in work faster and smarter. They are especially DevOps teams since they help them to increase successful deployments and reduce development time.

Vcs has a special kind of database in which they keeps track of every modification to the code .

2.3 Working with Git Local :

Following are the actions Git is able to Perform locally :-

- Creating local repository, adding files, and committing changes
- Viewing log and differences
- Staging changes as multiple changes
- Deleting and renaming files o Ignoring Files
- Undoing/redoing changes to the local copy and repository
- Cleaning the working copy

Git Commands:

git --version

Prints the Git suite model(version) that the git program got here from.

git –help

Prints the synopsis and a listing of the maximun used commands. If the option --all or -a is given then all to be had commands are printed.

2.4 Git Local Commands :

(i)Git init :

git init

this command is the first command that we use in Git. This command is used to creates a new blank repository and it is used to make existing project as a git project.

(ii)Git clone :

git clone <url>

this command can clone a repository from URL hosted location .

(iii)Git Config :

git config --global user.name "<firstname lastname>"

This command is used to set the user name of the user globally .

git config --global user.email "<valid-email>"

This command is used to set the email addfress of the user globally .

(iv)Git add :

git add <directory>

Stage all changes in <directory> for the next commit. Replace directory> with a <file> to change a specific file.

(v) Git Commit :

git commit -m *<message"

Commit the staged snapshot, but instead of launching a text editor, use <message> as the commit messag

(vi) Git Status :

git status

List which files are staged, unstaged, and untracked

(vii) Git log:

- git log -n - git log -n 3 → this command display only 3 commits .
- git log –oneline → this command show each commit to a single line.
- git log --author="" → Search for commits with a specific author. The argument may be a regular expression or normal string .
- git log --grep="" → Search for commits with a dedicated message that matches , which can be a regular expression or a string.
- git log - →this command display commits that include the specified file. This is an simple way to see the history of a specific file.

(vii)Git diff :

git diff HEAD → this command is used to Show difference between last commit and working directory.

git diff –cached → This command is used to Show difference between staged changes and last commit.

2.5 BRANCH & MERGE

git branch → This Command is used to get all the branch present in currently active bramch

git branch [branch-name] → This command will create the **branch [bramch-name]** locally in Git directory.

git checkout →this is used to switch to another branch in a repository.

git merge [branch] →this command is used to merge the current branch to specified branch's in the repository.

Working with Git remote(Github):

GitHub is a web-based Git repository hosting service, which offers all of the distributed revision control and source code management (SCM) functionality of Git as well as adding its own features.

- GitHub is how developers(peoples) build software
- With a network(community) of million people, developers can use, discover, and contribute to projects using a effective collaborative development workflow.
- GitHub is used as code hosting platform for version-control system and collaboration.
- GitHub helps you and others developers to work together on Same projects from anywhere.
- Nowdays GitHub is the most important online storage space of collaborative works that withinside the world.

Commands :

git remote add <name> <url>→ this command is used to Create a new connection in a remote repository. After adding connection to a remote, you can make shortcut for <url> as <name>.

git fetch <remote> <branch> → this command is used to Fetches a specific <branch>, from the repository And leave off <branch> to fetch all the remote refernces.

git pull <remote> → this command is used to Fetch the Particular remote's copy of workin(current) branch and directly merge it into the local copy of repository.

git push <remote> <branch> → this command is used to Push the specific branch to <remote>, along with important objects and commits ,Also Create named branch in the remote repository if it does not exist.

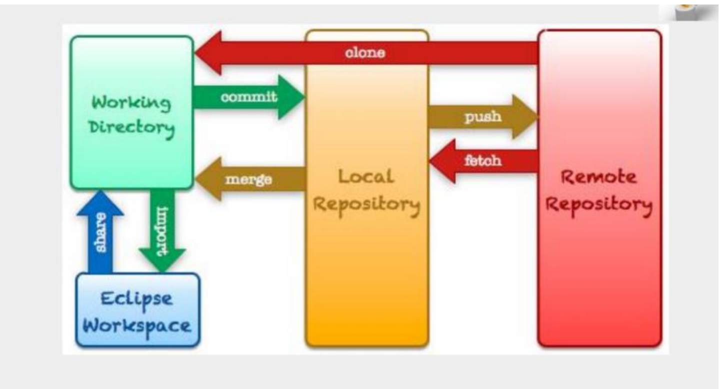


Fig. 9 Working with Git remote(Github)

2.1.4 JAVA:

What is Java?

Java is a programming language and a platform. Java is a high level, robust, object-oriented 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. The latest release of the Java Standard Edition is Java SE 8. With the advancement of Java and its widespread popularity, multiple configurations were built to suit various types of platforms. For example: J2EE for Enterprise Applications, J2ME for Mobile Applications.

Java is −

Object Oriented − In Java, everything is an Object. Java can be easily extended since it is based on the Object model.

Platform Independent − Unlike many other programming languages including C and C++, when Java is compiled, it is not compiled into platform specific machine, rather into platform independent byte code. This byte code is distributed over the web and interpreted by the Virtual Machine (JVM) on whichever platform it is being run on.

Simple − Java is designed to be easy to learn. If you understand the basic concept of OOP Java, it would be easy to master.

Secure − With Java's secure feature it enables to develop virus-free, tamper-free systems. Authentication techniques are based on public-key encryption.

Architecture-neutral − Java compiler generates an architecture-neutral object file format, which makes the compiled code executable on many processors, with the presence of Java runtime system.

Portable − Being architecture-neutral and having no implementation dependent aspects of the specification makes Java portable. Compiler in Java is written in ANSI C with a clean portability boundary, which is a POSIX subset.

Robust − Java makes an effort to eliminate error prone situations by emphasizing mainly on compile time error checking and runtime checking.

Multithreaded − With Java's multithreaded feature it is possible to write programs that can perform many tasks simultaneously. This design feature allows the developers to construct interactive applications that can run smoothly.

Interpreted − Java byte code is translated on the fly to native machine instructions and is not stored anywhere. The development process is more rapid and analytical since the linking is an incremental and light-weight process.

High Performance − With the use of Just-In-Time compilers, Java enables high performance.

Distributed − Java is designed for the distributed environment of the internet.

Dynamic − Java is considered to be more dynamic than C or C++ since it is designed to adapt to an evolving environment. Java programs can carry extensive amount of run-time information that can be used to verify and resolve accesses to objects on run-time.

OOPs in JAVA:

Object means a real-world entity such as a pen, chair, table, computer, watch, etc. Object-Oriented Programming is a methodology or paradigm to design a program using classes and objects. It simplifies software development and maintenance by providing some concepts.

- Object
- Class
- Inheritance
- Polymorphism
- Abstraction
- Encapsulation

Benefits of OOPs: -

- Real-**global(world)** programming
- Reusability of code
- Modularity of code
- Information hiding

1) Class

The class is one of the Basic concepts of OOPs which is a group of similar entities. It is only a logical component and not the physical entity.

Example: if you had a class called “Expensive Cars” it could have objects like Mercedes, BMW, Toyota, etc. Its properties(data) can be price or speed of these cars. While the methods may be performed with these cars are driving, reverse, braking etc.

2) Object

An object can be defined as an instance of a class, and there can be multiple instances of a class in a program.

For example – chair, bike, marker, pen, table, car, etc.

3) Inheritance

Inheritance is one of the Basic Concepts of OOPs in which one object acquires the properties and behaviors of the parent object. It’s creating a parent-child relationship between two classes.

4) Polymorphism

Polymorphism refers to one of the OOPs concepts in Java which is the ability of a variable, object or function to take on multiple forms.

5) Abstraction

Abstraction is one of the OOP Concepts which is an act of representing essential features without including background details.

Example: while driving a car, you do not have to be concerned with its internal working. Here you just need to concern about parts like steering wheel, Gears, accelerator, etc.

6) Encapsulation

Encapsulation is one of the best concepts of wrapping the data and code. In this OOPs concept, the variables of a class are always hidden from other classes.

Math Class

- Support **basic** mathematical functions.

- Can **now no longer** be instantiated.

- Round off Functions

  - Exponential Functions

  static double pow(double d1, double d2) :- This Is Explonential static function and return value in double.

  - Trigonometric Functions

  static double sin(double d) :- This Is Trignometric static function and give angle value in double.

  - random function generator

  static double random()

Wrapper Class

- Are used **to control** primitive values as objects.

- Are final.

- Objects of wrapper **instructions** are immutable

- static String toString(**kind** v) :- returns the string **similar to** the primitive **fee** of **kind handed** as argument.

int compareTo(WrapperType obj2)

type parseType(String s)

AutoBoxing and Autounboxing

5.4.4.1 Autoboxing

Auto Boxing is the process where the primitive data member automatically gets converted into its respective wrapper object. No need of explicit conversion by the programmer.

Integer intObj1 = new Integer(10); //boxing Integer intObj2 = 10; //auto boxing

5.4.4.2 Autounboxing

Auto unboxing is the process of automatically extracting the primitive value wrapped inside the object. No need of calling any method explicitly to fetch the primitive value.

int num1 = intObj1.intValue(); //unboxing int num2 = intObj2; //auto unboxing

Exception Handling

What is Exception Handling?

Exception Handling is a mechanism **to deal with** runtime **mistakes including** ClassNotFoundException, IOException, SQLException, RemoteException, etc.

5.5.1 Types of Java Exception

There are **in particular kinds of** exceptions: checked and unchecked. An **blunders** is **taken into consideration because the** unchecked exception. However, **in step with** Oracle, there are **3 kinds of** exceptions namely:

1. Checked Exception
2. Unchecked Exception
3. Error

Checked Exception

The **lessons** that **at once** inherit the Throwable **elegance besides** RuntimeException and Error are **referred to as** checked exceptions. For example, IOException, SQLException, etc. Checked exceptions are checked at compile-time.

Unchecked Exception

The **lessons(class)** that inherit the RuntimeException are **referred to as** unchecked exceptions. For **instance**, ArithmeticException, NullPointerException, ArrayIndexOutOfBoundsException, etc. Unchecked exceptions **aren't** checked at compile-time, **however they may be** checked at runtime.4.6.1.3 Error

Error Exception

Error is irrecoverable. Some **instance** of **mistakes** are OutOfMemoryError, VirtualMachineError, AssertionError etc.

Java Exception Keywords

Java **offers 5 key phrases which can be** used **to deal with** the exception. The following table describes each.

Keyword	**Description**
try	try key-word is used to specify a block **wherein** we **must location** an exception code. It **way** we **cannot** use **attempt** block alone. The **attempt** block **have to** be **observed through both seize** or **sooner or later**.
catch	Catch is used **to deal with** the exception. It **have to** be preceded **through attempt** block **this means that** we **cannot** use **seize** block alone. It **may be observed through sooner or later** block later.
finally	finally is used to execute the **important** code of the program. It is **performed whether or not** an exception is **treated** or not.
throw	Throw key-word is used to throw an exception.
throws	Throws **key-word** is used to **claim** exceptions. It specifies that there **can also additionally arise** an exception **withinside the technique**. It **does not** throw an exception. It is **usually** used with **technique** signature.

``2.1.5 MAVEN

What is MAVEN ?

Apache Maven is a software project management and comprehension tool. Based on the concept of a project object model (POM), Maven can manage a project's build, reporting and documentation from a central piece of information.

Maven was created to simplify Jakarta Tribune project building processes. Many of the projects had slightly different ANT files, so Apache developed Maven to handle building multiple projects together, including publishing project information, facilitating team collaboration, deploying projects, and sharing JARS among several projects.

Objective:

Maven’s purpose is to provide developers with:

- A comprehensive, maintainable, reusable, and simple model for projects.
- A set of tools and plug-ins that can interact with the declarative model.

Features:

Maven is loaded with many valuable and useful features, which goes a long way towards explaining its popularity. Here are some of Maven’s more noteworthy features:

- A huge, continuously growing repository of user libraries
- The ability to set up projects easily, using best practices
- Dependency management, featuring automatic updating
- Backwards compatible with previous versions
- Strong error and integrity reporting
- Automatic parent versioning
- Ensures consistent usage across all projects
- It’s extensible, and you can easily write plug-ins using scripting languages or Java.

Broader pictures

- Lifecycle - broadest unit

- Phase - A stage in the lifecycle.

- Goal - A plugin goal represents a specific task

Lifecycles in maven

• default - handles your project deployment

• clean - handles project cleaning

• site - handles the creation of your project's site documentation

Default Lifecycle phases

• **validate** - validate the project is correct and all necessary information is available

• **compile** - compile the source code of the project

• **test** - test the compiled source code using a suitable unit testing framework. These tests should not require the code be packaged or deployed

• **package -** take the compiled code and package it in its distributable format, such as a JAR.

• **verify** - run any checks on results of integration tests to ensure quality criteria are met

• i**nstall** - install the package into the local repository, for use as a dependency in other projects locally

• **deploy** - done in the build environment, copies the final package to the remote repository for sharing with other developers and projects.

Maven Coordinates

- The Archetype plugin created a assignment(project) with a document named pom.xml.
- This is the Project Object Model (POM), a declarative description of a assignment
- Maven coordinates outline a hard and fast of identifiers which may be used to uniquely become aware of a assignment .
    - groupId
    - artifactId
    - Version (Snapshot/1.x/2.x)
    - Packaging (jar/war/ear/pom)
- Packaging – (Jar/war/ear/pom) Represent the package for the Project in which we work.
- artifactId – Represent the name of the Project .
- groupId – it is the combination of packaging and artifactid to get unique groupis.
- Version – represent the current version of the project.

Maven POM

• Stands for Project Object Model

• Describes a assignment (project)

- Name and Version

- Artifact Type

- Dependencies

- Plugins

- Profiles

• By default They uses Xml file.

Project Object Model

• Maven projects, dependencies, builds, artifacts: all **of those** are **items** to be modeled and **defined withinside the** POM **report**.

• The POM tells Maven what **type of project it is.** Also managing modifying default behaviour to bring out output from source.

• Maven project is **described via way of means of** the presence of a pom.xml

• It is a descriptive an**nouncement** of a project for Maven; **it's far** the figurative “map” that Maven **desires** to understand

• Somewhat analogous to the **construct report** in ant.

It should be noted that there should be a single POM file for each project.

- All POM files require the project element and three mandatory fields: groupId, artifactId, version.
- Projects notation in repository is groupId:artifactId:version.

Advantages of Maven

- Helps manage all the processes, such as building, documentation, releasing, and distribution in project management
- Simplifies the process of project building
- Increases the performance of the project and the building process
- The task of downloading Jar files and other dependencies is done automatically
- Provides easy access to all the required information
- Makes it easy for the developer to build a project in different environments without worrying about the dependencies, processes, etc.
- In Maven, it’s easy to add new dependencies by writing the dependency code in the pom file

Disadvantages of Maven

- Maven requires installation in the working system and the Maven plug-in for the IDE
- If the Maven code for an existing dependency is unavailable, you cannot add that dependency using Maven itself
- Some sources claim that Maven is slow

2.1.6 SPRING

Spring framework is an open source Java platform that provides comprehensive infrastructure support for developing robust Java applications very easily and very rapidly. Spring framework was initially written by Rod Johnson and was first released under the Apache 2.0 license in June 2003.

Applications of Spring

Following is the list of few of the great benefits of using Spring Framework −

- POJO Based - Spring enables developers to develop enterprise-class applications using POJOs. The benefit of using only POJOs is that you do not need an EJB container product such as an application server but you have the option of using only a robust servlet container such as Tomcat or some commercial product.
- Modular - Spring is organized in a modular fashion. Even though the number of packages and classes are substantial, you have to worry only about the ones you need and ignore the rest.
- Integration with existing frameworks - Spring does not reinvent the wheel, instead it truly makes use of some of the existing technologies like several ORM frameworks, logging frameworks, JEE, Quartz and JDK timers, and other view technologies.
- Testablity - Testing an application written with Spring is simple because environment-dependent code is moved into this framework. Furthermore, by using JavaBeanstyle POJOs, it becomes easier to use dependency injection for injecting test data.
- Web MVC - Spring's web framework is a well-designed web MVC framework, which provides a great alternative to web frameworks such as Struts or other over-engineered or less popular web frameworks.
- Central Exception Handling - Spring provides a convenient API to translate technology-specific exceptions (thrown by JDBC, Hibernate, or JDO, for example) into consistent, unchecked exceptions.
- Lightweight - Lightweight IoC containers tend to be lightweight, especially when compared to EJB containers, for example. This is beneficial for developing and deploying applications on computers with limited memory and CPU resources.
- Transaction management - Spring provides a consistent transaction management interface that can scale down to a local transaction (using a single database, for example) and scale up to global transactions (using JTA, for example).

2.1.7 Front-end Technologies:

- HTML
- CSS
- JAVASCRIPT

What is HTML?

HTML stands for Hyper Text Markup Language. HTML helps you structure your page into elements such as paragraphs, sections, headings, navigation bars, and so on. **HTML** provides the *basic structure* of sites, which is enhanced and modified by other technologies like CSS and JavaScript.

What is CSS ?

CSS stands for Cascading Style Sheets, and you use it to improve the appearance of a web page. By adding thoughtful CSS styles, you make your page more attractive and pleasant for the end user to view and use, so CSS is a design language that you use to make your web page look nice and presentable.

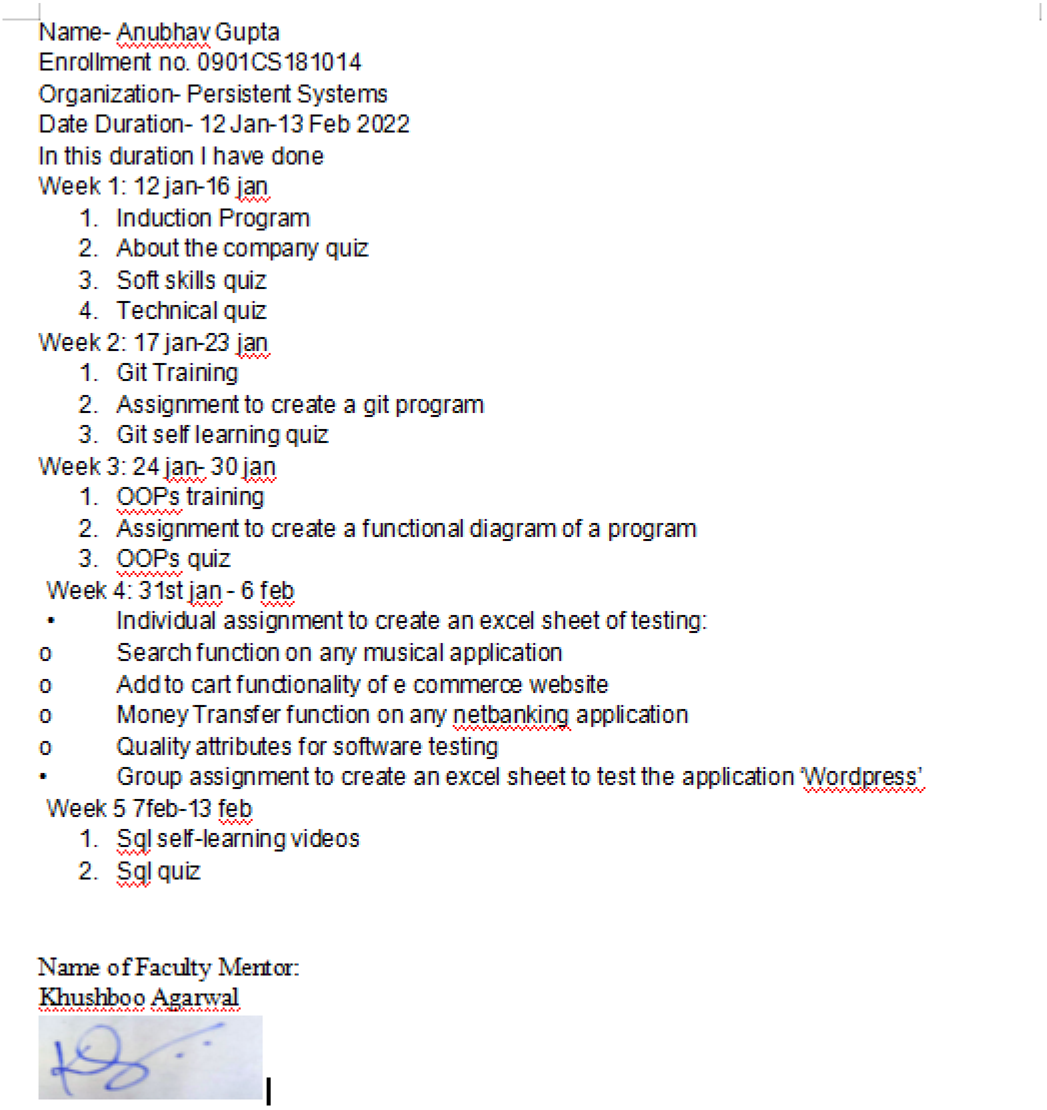
What is Javascript ?

JavaScript is used to control the *behavior* of different elements.

CONCLUSION

The experience of getting internship in Persistent Systems allowed me to develop and learn new technologies like SQL, GIT, HTML, CSS etc. More than this we at Persistent Systems get into non-tech activities like Experiential Learning Championship(ELC) where we have to participate as team. I have also develop the quality of communication and work as team member and how to convince yourself to your team member and complete the given activity.

FPR-1



Name- Anubhav Gupta
Enrollment no. 0901CS181014
Organization- Persistent Systems
Date Duration- 12 Jan-13 Feb 2022
In this duration I have done

Week 1: 12 jan-16 jan

1. Induction Program
2. About the company quiz
3. Soft skills quiz
4. Technical quiz

Week 2: 17 jan-23 jan

1. Git Training
2. Assignment to create a git program
3. Git self learning quiz

Week 3: 24 jan- 30 jan

1. OOPs training
2. Assignment to create a functional diagram of a program
3. OOPs quiz

Week 4: 31st jan - 6 feb

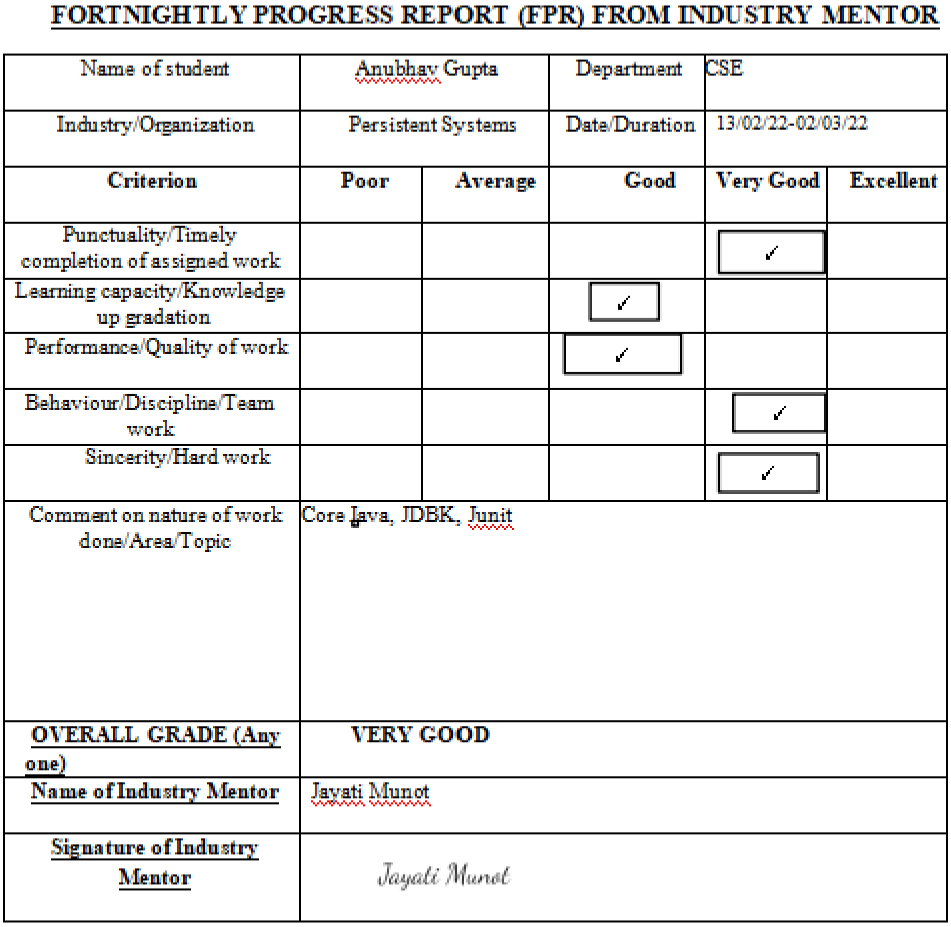
- Individual assignment to create an excel sheet of testing:
  - Search function on any musical application
  - Add to cart functionality of e commerce website
  - Money Transfer function on any netbanking application
  - Quality attributes for software testing
- Group assignment to create an excel sheet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

Week 5 7feb-13 feb

1. Sql self-learning videos
2. Sql quiz

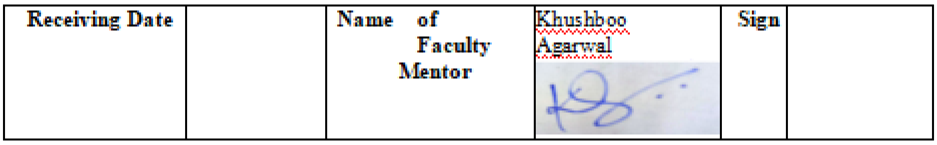
Name of Faculty Mentor:
Khushboo Agarwal

FPR-2



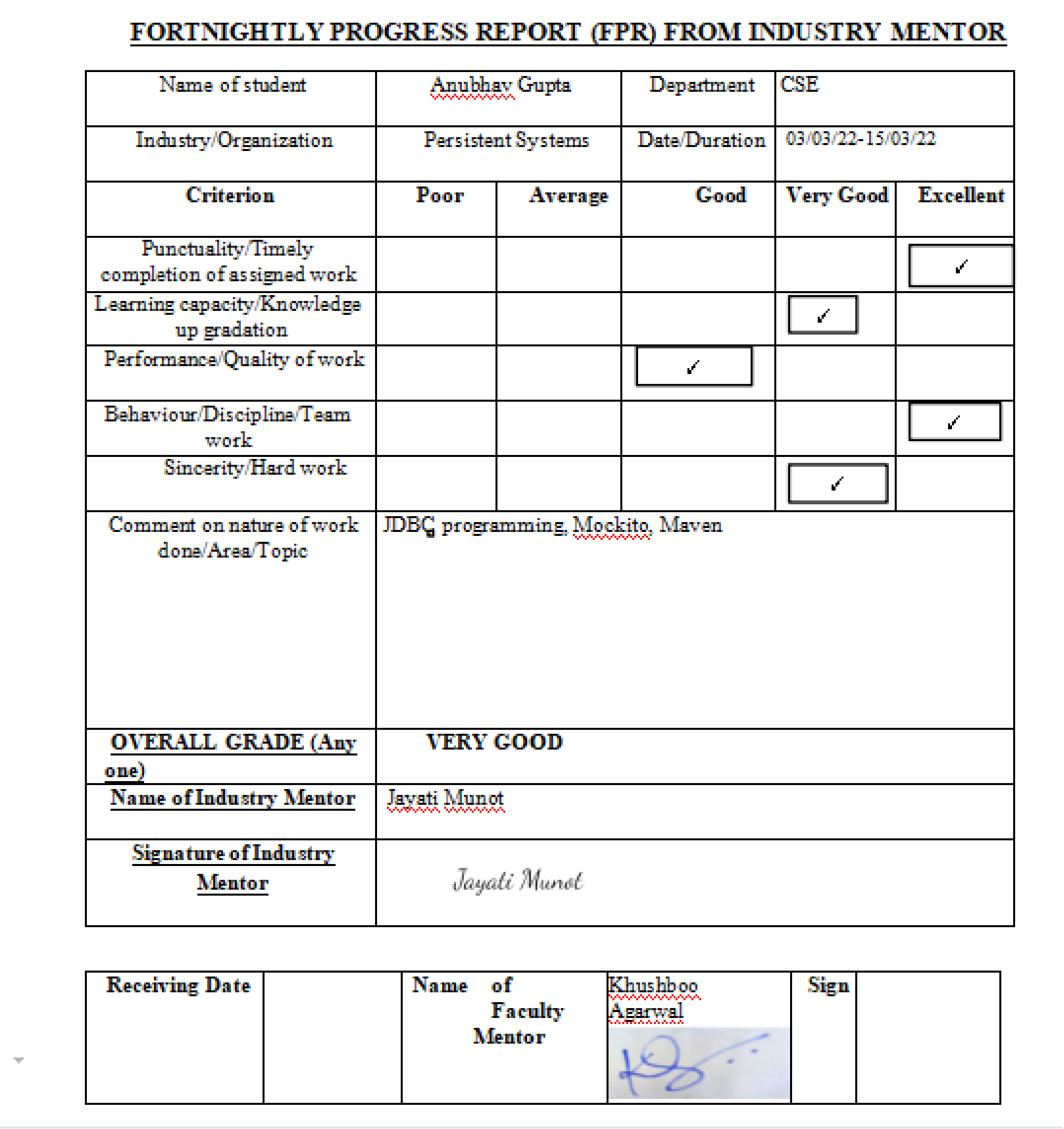
FORTNIGHTLY PROGRESS REPORT (FPR) FROM INDUSTRY MENTOR

Name of student	Anubhav Gupta		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**	Khushboo Agarwal	**Sign**	

FPR-3

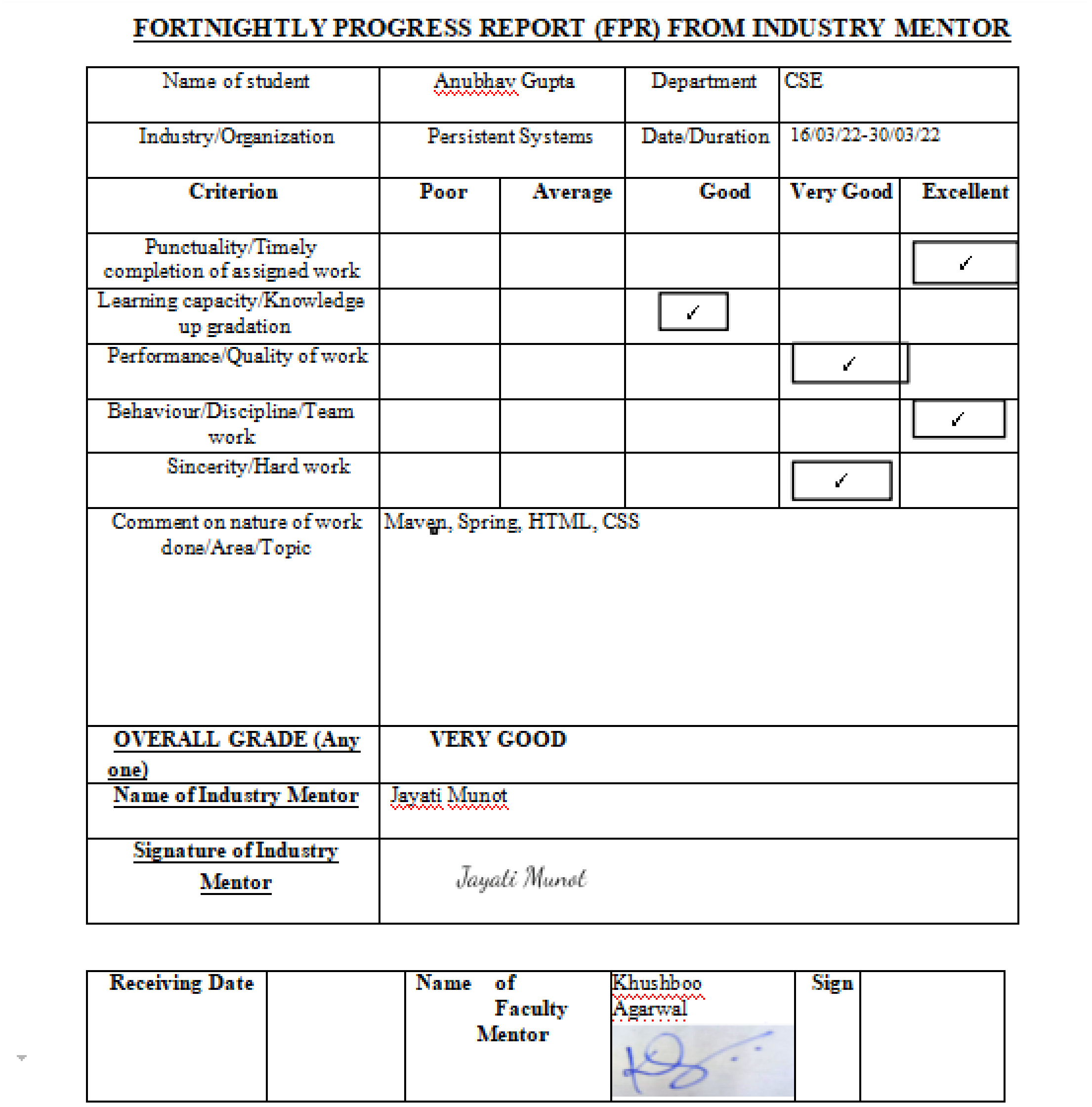


FORTNIGHTLY PROGRESS REPORT (FPR) FROM INDUSTRY MENTOR

Name of student	Anubhav Gupta		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**	Khushboo Agarwal	**Sign**	

FPR-4

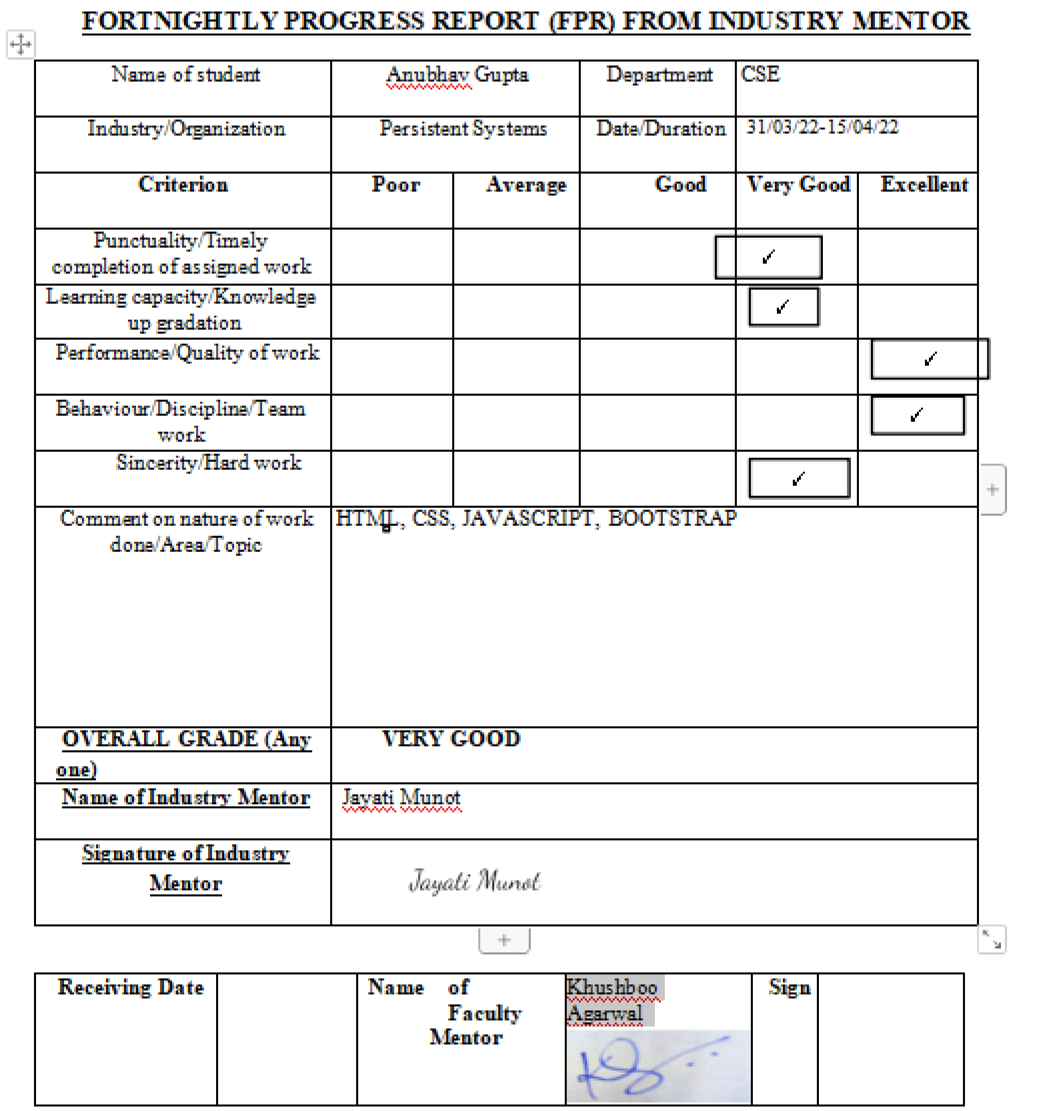


FORTNIGHTLY PROGRESS REPORT (FPR) FROM INDUSTRY MENTOR

Name of student	Anubhav Gupta		Department	CSE	
Industry/Organization	Persistent Systems		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**	Khushboo Agarwal	**Sign**	

FPR-5

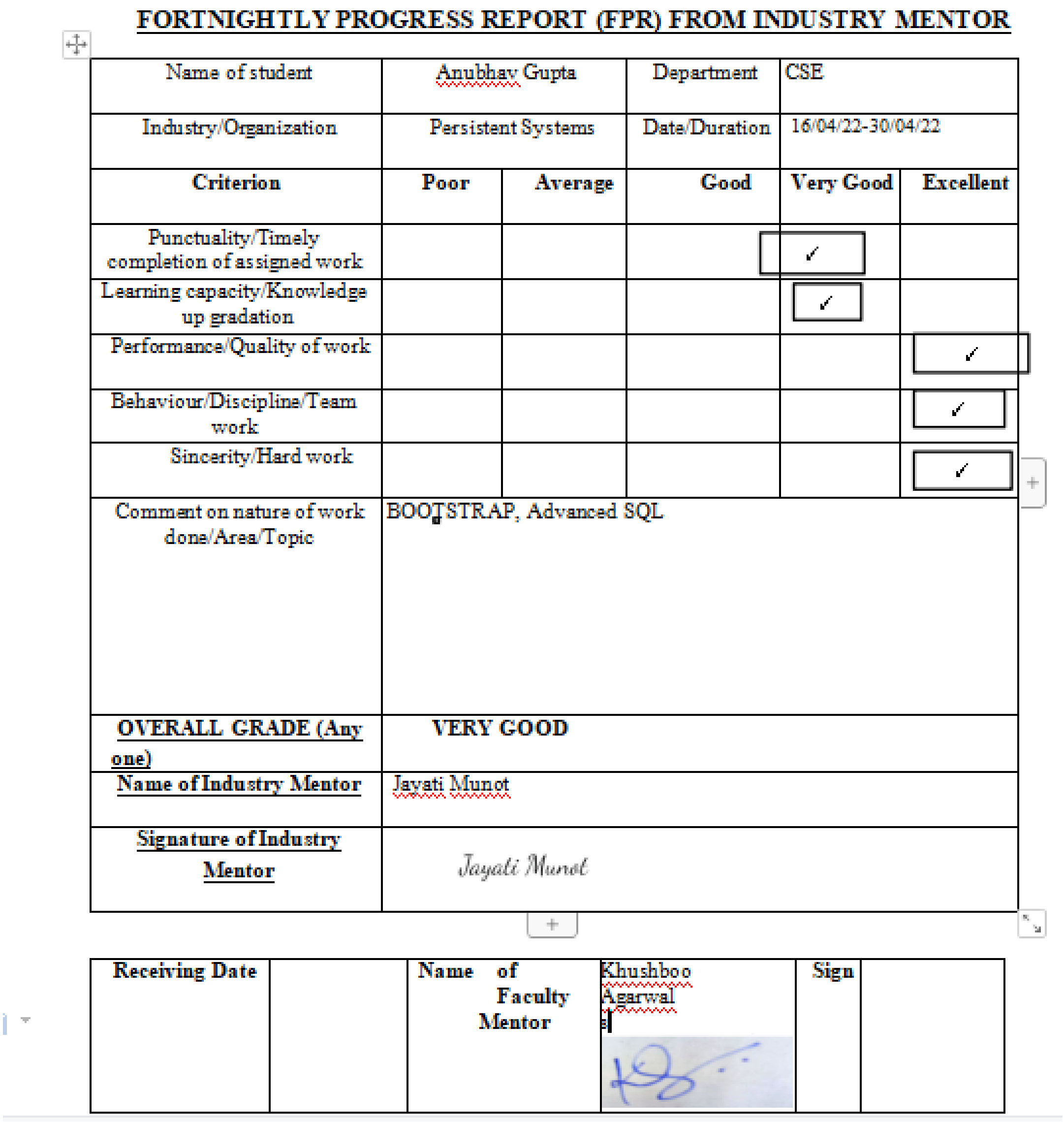


FORTNIGHTLY PROGRESS REPORT (FPR) FROM INDUSTRY MENTOR

Name of student	Anubhav Gupta		Department	CSE	
Industry/Organization	Persistent Systems		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**	Khushboo Agarwal	**Sign**	

FPR-6

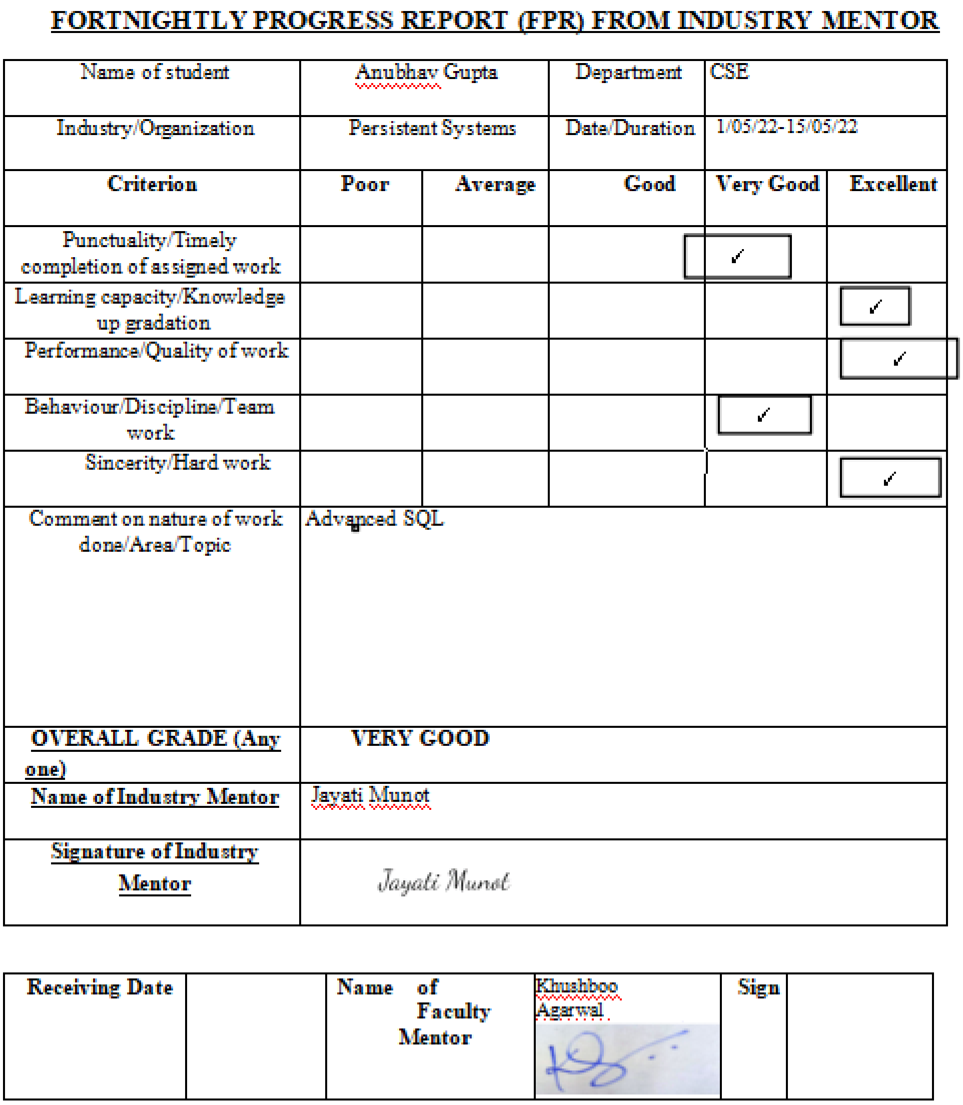


FORTNIGHTLY PROGRESS REPORT (FPR) FROM INDUSTRY MENTOR

Name of student	Anubhav Gupta		Department	CSE	
Industry/Organization	Persistent Systems		Date/Duration	16/04/22-30/04/22	
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	BOOTSTRAP, 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**	Khushboo Agarwal	**Sign**	

FPR-7

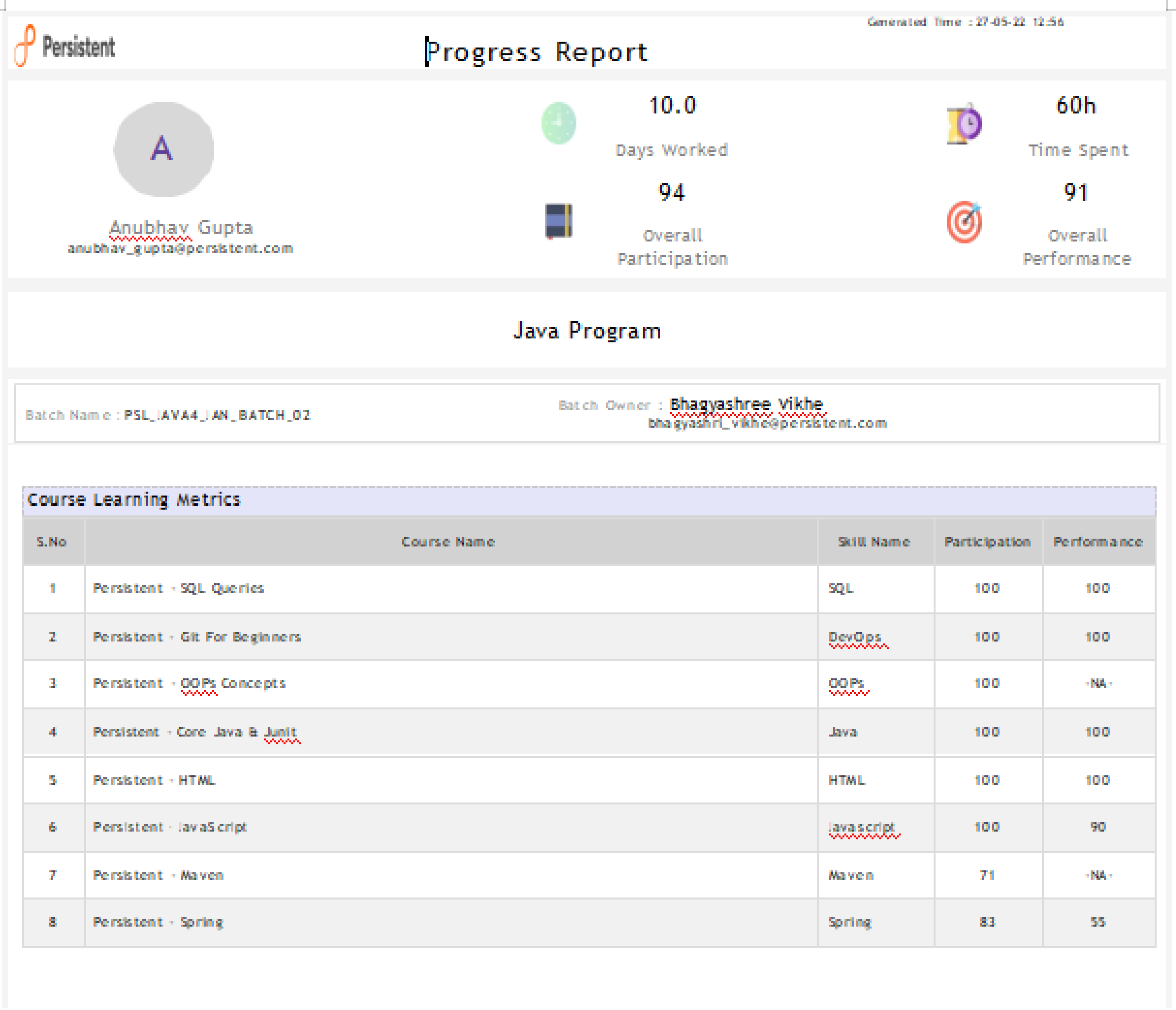


FORTNIGHTLY PROGRESS REPORT (FPR) FROM INDUSTRY MENTOR

Name of student	Anubhav Gupta		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	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**	Khushboo Agarwal	**Sign**	

Progress Report:



Generated Time : 27-05-22 12:56

Progress Report

A

Anubhav Gupta
anubhav_gupta@persistent.com

10.0
Days Worked

60h
Time Spent

94
Overall Participation

91
Overall Performance

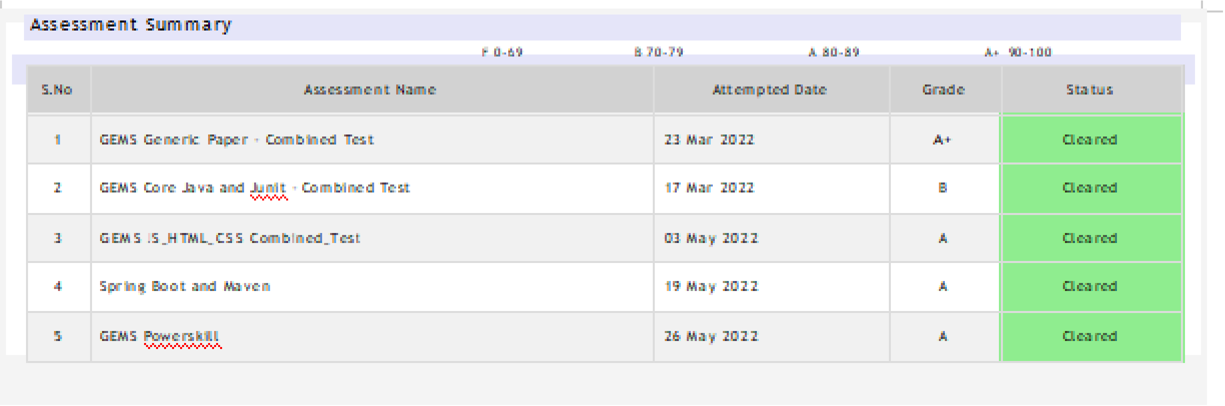
Java Program

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	100	100
2	Persistent - Git For Beginners	DevOps	100	100
3	Persistent - OOPs Concepts	OOPs	100	-NA-
4	Persistent - Core Java & Junit	Java	100	100
5	Persistent - HTML	HTML	100	100
6	Persistent - JavaScript	javascript	100	90
7	Persistent - Maven	Maven	71	-NA-
8	Persistent - Spring	Spring	83	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	23 Mar 2022	A+	Cleared
2	GEMS Core Java and Junit - Combined Test	17 Mar 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	26 May 2022	A	Cleared

REFERENCES

1. Persistent material
2. Udemy Material
3. https://en.wikipedia.org/wiki/Git
4. Java OOPs Concepts - Javatpoint
5. https://online.visual-paradigm.com/diagrams/solutions/free-uml-tool
6. https://www.javatpoint.com/what-is-core-java
7. https://www.javatpoint.com/spring-mvc-model-interface
8. Spring tutorial examples eclipse (w3schools.blog)
9. https://www.javatpoint.com/maven-tutorial