Automation of measurement metrics

A Project-II Report

Submitted in partial fulfillment of requirement of the Degree of

# BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE & ENGINEERING

BY

**Suhani Neema**

### EN18CS301272

Under the Guidance of

**Professor Mr. Prasanna Kapse**



**Department of Computer Science & Engineering Faculty of Engineering**

**MEDI-CAPS UNIVERSITY, INDORE- 453331**

Automation of measurement metrics

A Project-II Report

Submitted in partial fulfillment of requirement of the Degree of

# BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE & ENGINEERING

BY

**Suhani Neema**

### EN18CS301272

Under the Guidance of

**Professor Mr. Prasanna Kapse**



**Department of Computer Science & Engineering Faculty of Engineering**

**MEDI-CAPS UNIVERSITY, INDORE- 453331**

**April, 2022**

# Report Approval

#### The project work **“Automation of measurement metrics”** is hereby approved as a creditable study of an engineering/computer application subject carried out and presented in a manner satisfactory to warrant its acceptance as prerequisite for the Degree for which it has been submitted.

It is to be understood that by this approval the undersigned do not endorse or approved any statement made, opinion expressed, or conclusion drawn there in; but approve the “Project Report” only for the purpose for which it has been submitted.

Internal Examiner Name: Designation Affiliation

External Examiner Name: Designation Affiliation

# Declaration

#### I/We hereby declare that the project entitled **“Automation of measurement metrics”** submitted in partial fulfillment for the award of the degree of Bachelor of Technology in ‘Computer Science and Engineering’ completed under the supervision of **Mr. Prasanna Kapse,** Faculty of Engineering, Medi-Caps University Indore is an authentic work.

Further, I/we declare that the content of this Project work, in full or in parts, have neither been taken from any other source nor have been submitted to any other Institute or University for the award of any degree or diploma.

**Suhani Neema EN18CS301272**

**Certificate**

I, **Prasanna Kapse** certify that the project entitled “**Automation of measurement metrics”** submitted in partial fulfillment for the award of the degree of Bachelor of Technology by **Suhani Neema** is the record carried out by him/them under my/our guidance and that the work has not formed the basis of award of any other degree elsewhere.

Mr. Prasanna Kapse

Computer Science

#### Medi-Caps University, Indore

Mr. Karthik\_Vg

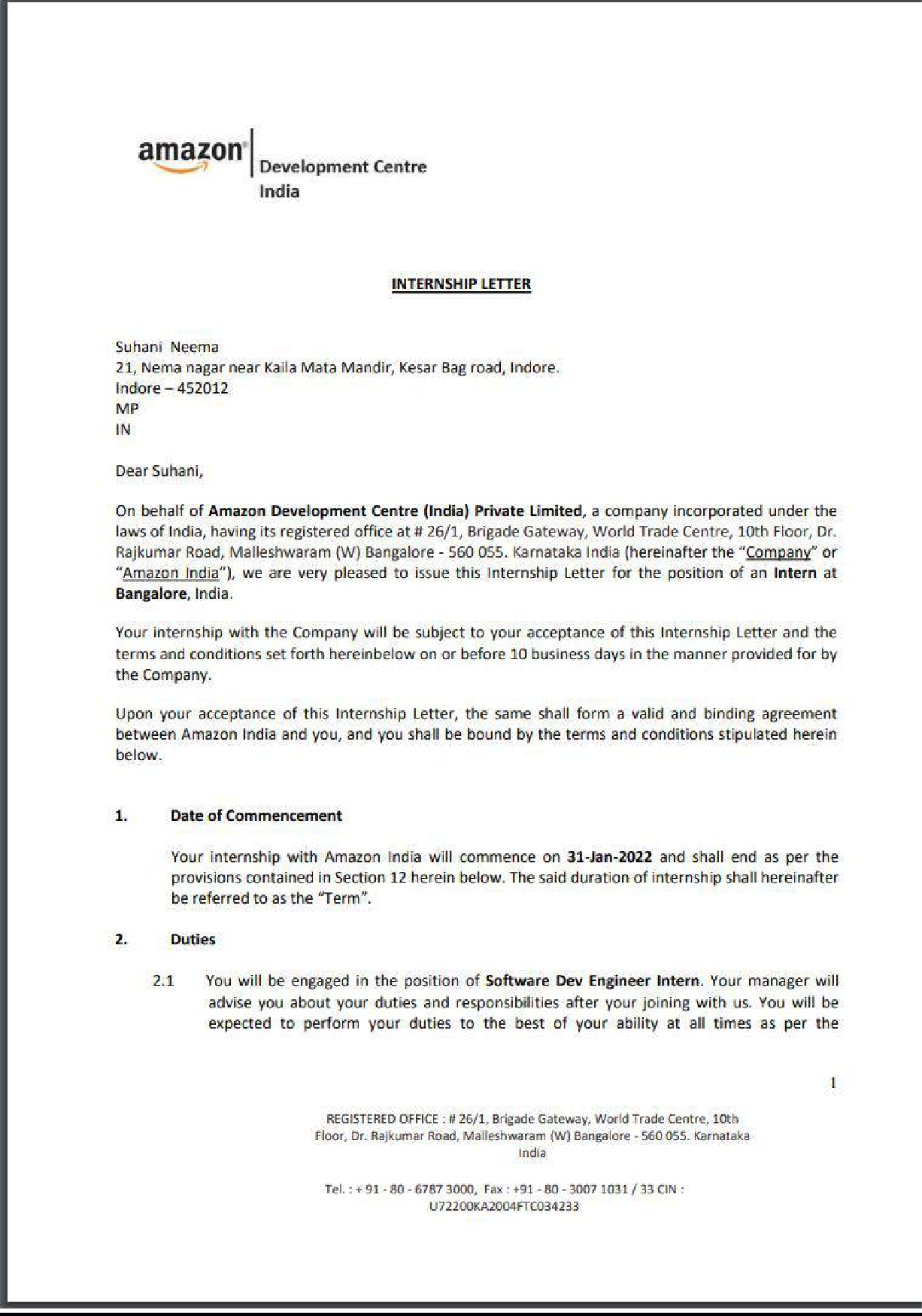
Mentor

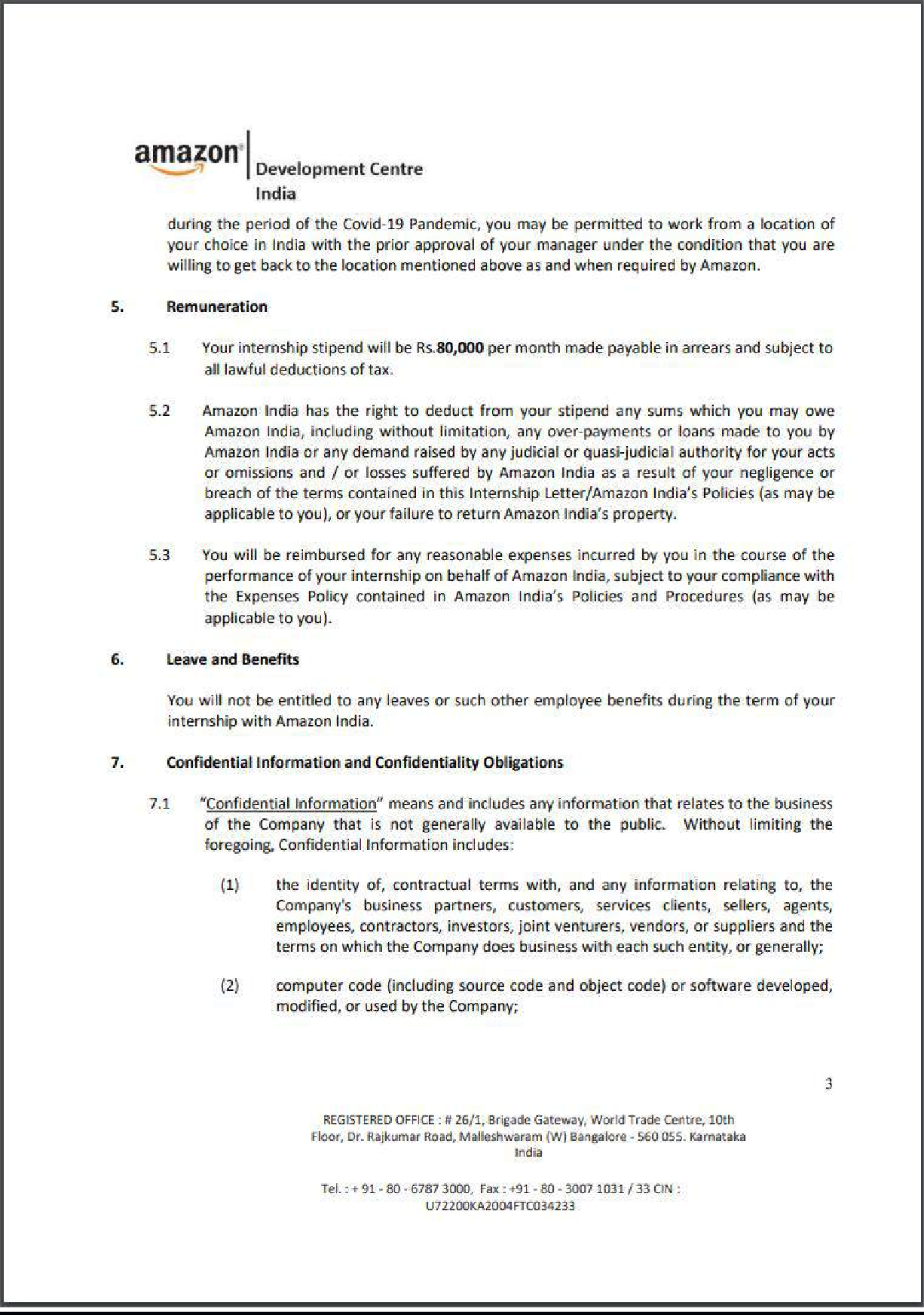
#### Amazon Development Center.

Dr. Pramod S. Nair Head of the Department

Computer Science & Engineering Medi-Caps University, Indore

# Offer Letter of the Project work-II/Internship





**Completion certificate/Letter**

**Acknowledgements**

I would like to express my deepest gratitude to Honorable Chancellor, **Shri R C Mittal,** who has provided me with every facility to successfully carry out this project, and my profound indebtedness to **Prof. (Dr.) Dileep K Patnayak,** Vice Chancellor, Medi-Caps University, whose unfailing support and enthusiasm has always boosted up my morale. I also thank **Prof. (Dr.) D K Panda,** Pro Vice Chancellor, **Dr. Suresh Jain,** Dean Faculty of Engineering, Medi-Caps University, for giving me a chance to work on this project. I would also like to thank my Head of the Department **Dr. Pramod S. Nair** for his continuous encouragement for betterment of the project.

I express my heartfelt gratitude to my **External Guide, Mr. Karthik\_Vg**, mentor, Amazon Development Center, India as well as to my Internal Guide, Mr. Prasanna Kapse, Professor, Department of Computer Science and Engineering, MU, whose continuous help and support, helped me to accomplish the components of project.

It is their help and support, due to which we became able to complete the design and technical report. Without their support this report would not have been possible.

### Suhani Neema

B.Tech. IV Year

Department of Computer Science & Engineering Faculty of Engineering

Medi-Caps University, Indore

# ABSTRACT

As a part of my under-graduate program, I joined Amazon Development Center, India, on 31 January 2022 as a Software Development Intern . In the initial training of 2 weeks, I went through the understanding of Amazon internal system work-flow and later on in the upcoming weeks of my internship training I had to learn and simultaneously implement the technologies.

As a part of my internship project I’ve automated the process of generating schedule sheet for the onboarding of new HVA into HVA 2.0 production Systems and also worked upon creating DAO layer to interact with DynamoDB .

Amazon Development Center, in India work on complex technology challenges to innovate and create efficient solutions that enable various Amazon businesses, including Amazon websites across the world as well as support Payments, Transportation, and Digital products and services like the Kindle family of tablets, e-readers and the store.

# Table of Contents

|  |  |  |
| --- | --- | --- |
|  |  | **Page No.** |
|  |  |  |
|  | Report Approval | ii |
|  | Declaration | iii |
|  | Certificate | iv |
|  | Offer Letter of the Project work-II/Internship | v |
|  | Completion letter/certificate | vi |
|  | Acknowledgement | vii |
|  | Abstract | viii |
|  | Table of Contents | ix |
| Chapter 1 | Introduction | 2 |
|  | 1.1 Introduction | 2 |
|  | 1.2 Components of Initial Training | 2 |
|  | 1.3 overview of Internship Tasks. | 3 |
| Chapter 2 | Builder Tools Workshop | 4 |
|  | 2.1 Getting Started | 4 |
|  | 2.2 Building services, Code changes. | 6 |
| Chapter 3 | Operating System | 7 |
|  | 3.1 Operating System – Introduction | 8 |
|  | 3.2 LINUX CLI and Commands | 8 |
| Chapter 4 | Version Control System | 11 |
|  | 4.1 Version Control System -Introduction | 11 |
|  | 4.2 Git | 11 |
| Chapter 5 | IDE | 14 |
|  | 5.1 Introduction to Project setup. | 14 |
| Chapter 6 | System requirement analysis | 16 |
|  | 6.1 Information Gathering | 16 |
|  | 6.2 System Feasibility | 16 |
|  | 6.3 Platform Specification | 17 |
| Chapter 7 | In-depth technological analysis | 18 |
|  | 7.1 Databases | 18 |
|  | 7.2 Frameworks | 18 |
|  | 7.3 Languages | 19 |
| Chapter 8 | Testing | 21 |
|  | 8.1 Unit Testing | 21 |
|  | 8.2 Integration Testing | 21 |
| Chapter 9 | Results and Discussions | 22 |
| Chapter 10 | Conclusions | 23 |

**Chapter-1**

## Internal Training

* 1. **Introduction**

“Work Hard. Have fun. Make History.” The training team of Amazon Development Center, India are a set of working professionals and a vibrant tech group, whose pursuit is progress for people everywhere. That's why we take a closer look at things, learn them, spread them and ask questions and think ahead.

I work under CBA org in Amazon, which stands for Customer Behavior Analytics.

#### About CBA:

As a pioneer in commerce, Amazon leads the industry in delivering new and exceptional shopping experiences. Our relentless focus on customers drives us to innovate and delight shoppers, no matter where they buy. The Customer Behavior Analytics (CBA) organization owns Amazon’s insights pipeline, from data collection to deep analytics. We aspire to be the place where Amazon teams come for answers, a trusted source for data and insights that empower our systems and business leaders to make better decisions. Our outputs shape Amazon product and marketing teams’ decisions and thus how Amazon customers see, use, and value their experience.

Working at amazon always feels like **“It’s Day One”**

This phrase always helps us to be motivated, to learn more, share your knowledge among your peers. This helps us to thrive for the best.

The initial training was of 2 weeks. Which comprised of several important topics that are required for the understanding the workflow of software development and other required topics.

## Components of the Initial Training

The initial training was to go through their Builder Tools Workshop, and later on it was work specific:

* + 1. Builder Tools Workshop.
    2. Operating Systems.
    3. Frameworks.
    4. Database.
    5. Technology Stack

Other than these I also learned and worked on Scala, Python, Spark, S3, Zeppelin, pandas.

## Overview of internship tasks

* + 1. **Schedule Sheet Generation**

In this case, I was assigned to automate the process of generating the schedule sheet, and for this task I have been first assigned to search for the appropriate framework in JAVA that can help us to automate the generation of schedule sheet.

I tried to break this tasks into sub components and those are:

1. Try to find the appropriate the framework for it.
   1. I was able to find two libraries, J-Excel and Apache-Poi, later on gathered more information, whether we need to automate .xlsx file or.xls on that basis I chose, Apache-Poi, because it supports that file.
   2. Whereas, in J-Excel, we are not allowed to use to .xlsx files.
2. After confirming the framework, I went on finding the internal library that we use in Amazon for Apache-Poi, and hence find it built it into the version set.
3. After this, started working on the code Part, where I broke it down into sub-components again:
   1. What I have to Do?
   2. How will I do it?
      1. First searched it how can I fill the files with the help of Apache-Poi?
      2. Find it implemented it.
      3. Next tried to implemented it in my kind of usecase.
      4. Later on found the way how I can create another rows and copy some of the data into it.
      5. On the basis of these 4 points, I was able to write the code.
4. Unit Testing.

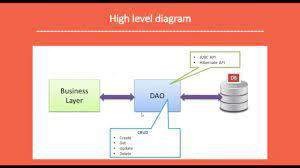
## DAO interface to interact with DynamoDB

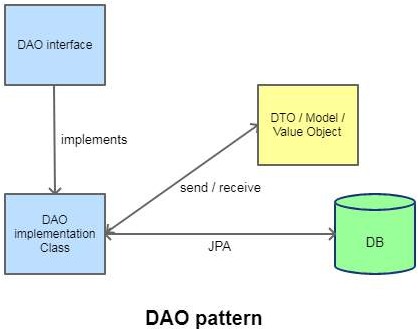
In this part of task, I was assigned to write the DAO and Ramp up on Dynamo DB. DAO:

The Data Access Object (DAO) pattern is a structural pattern that allows us to isolate the application/business layer from the persistence layer (usually a relational database but could be any other persistence mechanism) using an abstract API.

The API hides from the application all the complexity of performing CRUD operations in the underlying storage mechanism. This permits both layers to evolve separately without knowing anything about each other.

On getting some part of understanding interacted with the team and get the insights on what all things we could have in our DAO layers, on finalizing this started writing DAO layer and then implemented the same to interact with DynamoDB.





# Chapter-2

## Builder Tool Workshop

* 1. **Getting Started**

As a new builder in Amazon we have received a development kit which requires some certain set of permissions and have some certain rules.

With that we had make sure to meet the certain Amazon internal prerequisites to access the source and tools, need to setup our development environment and cloud desktop.

**Development desktop**: It is the place where we develop in our local system test it, run it, fixes bug. It could be either of macOs, Windows, Ubuntu. I used macOs.

**Cloud Desktop**: A cloud desktop is a linux instance that contains all the tools that we need to develop software.

**Builder System:** Amazon has its internal development system where we build our package and build system.

## Building services, Code changes.

Here, we understand team responsibility to build, maintain and supports the web services.

To write a code, we learnt that we will always be creating a workspace where it contains all of the packages that we are working with.

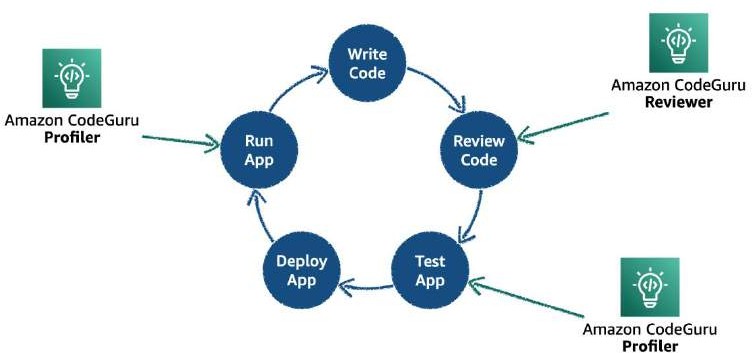
So once the package has been created, we look for the dependencies that we will be needing in our use-case, we will search the dependencies in amazon vast code bases.

And then we build our version set and packages using those dependencies, amazon has a internal tool for that.

To add the dependencies related to our use-case, we need to add the dependencies in our Config file.

Then we need to sync the dependencies, workspace and version set that we are using. Once it is done we need to code, make changes and run it build it and test it.

Once we are done with we use git commands to make the changes available to the reviewers and allow them to review the code.



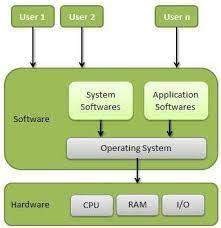
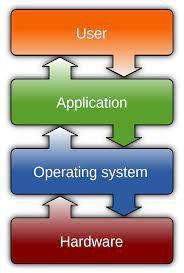
# Chapter-3

## Operating System

* 1. **Operating System – Introduction**

An operating system is the core set of software on a device that keeps everything together. The operating system manages the communication between your software and your hardware.

Without the operating system (often referred to as the “OS”), the software wouldn’t function. The operating system sits in between the applications you run and the hardware, using the hardware drivers as the interface between the two.





### LINUX Operating System

Linux is the best-known and most-used open-source operating system. As an operating system, Linux is software that sits underneath all of the other software on a computer, receiving requests from those programs and relaying these requests to the computer’s hardware. ❖ HOW LINUX IS DIFFERENT? It is an open-source OS which gives a great advantage to the programmers as they can design their own custom operating system. It gives you a lot of

options for programs having some different features so you can choose according to your needs. Above all, you don't have to pay for software and server licensing to install Linux, it’s absolutely free and you can install it on as many computers as you want. It’s a completely trouble-free operating system and don't have an issue with viruses, malware and slowing down your computer. Linux is Mainly used in servers. About 90% of the internet is powered by Linux servers. This is because Linux is fast, secure, and free! The main problem of using Windows servers is their cost. This is solved by using Linux servers. The OS that runs in about 80% of the smartphones in the world, Android, is also made from the Linux kernel. Most of the viruses in the world run on Windows, but not on Linux!

* Structure of Linux
* Hardware − Hardware consists of all physical devices attached to the System. For example: Hard disk drive, RAM, Motherboard, CPU etc.
* Kernel − Kernel is the core component for any (Linux) operating system which directly

interacts with the hardware.

* Shell – Shell is the interface which takes input from Users and sends instructions to the Kernel, Also takes the output from Kernel and send the result back to output shell.
* Applications − These are the utility programs which runs on Shell. This can be any

application like Your web browser, media player, text editor etc

* LINUX Distributions:

There are on an average six hundred Linux distributors providing different features. But popular ones are:

* Ubuntu
* Linux
* Mint
* Debian

### 3.2 LINUX CLI and Commands

* Linux Shell or “Terminal”:

A shell is a program that receives commands from the user and gives it to the OS to process, and it shows the output. Linux's shell is its main part. Its distros come in GUI (graphical user interface), but basically, Linux has a CLI (command line interface). In this tutorial, we are going to cover the basic commands that we use in the shell of Linux. To open the terminal, press Ctrl+Alt+T in Ubuntu, or press Alt+F2, type in gnome-terminal, and press enter. In Raspberry Pi, type in lxterminal. There is also a GUI way of taking it, but this is better!

* Basic Linux Commands

1. pwd — When you first open the terminal, you are in the home directory of your user. To know which directory you are in, you can use the “pwd” command. It gives us the absolute path, which means the path that starts from the root. The root is the base of the Linux file system. It is denoted by a forward slash (/). The user directory is usually something like "/home/username".
2. ls — Use the "ls" command to know what files are in the directory you are in. You can see

all the hidden files by using the command “ls -a”.

1. cd — Use the "cd" command to go to a directory. For example, if you are in the home folder, and you want to go to the downloads folder, then you can type in “cd Downloads”. Remember, this command is case sensitive, and you have to type in the name of the folder exactly as it is.
2. mkdir & rmdir — Use the mkdir command when you need to create a folder or a directory. For example, if you want to make a directory called “DIY”, then you can type “mkdir DIY”. Remember, as told before, if you want to create a directory named “DIY Hacking”, then you can type “mkdir DIY\ Hacking”. Use rmdir to delete a directory. But rmdir can only be used to delete an empty directory. To delete a directory containing files, use rm.
3. rm - Use the rm command to delete files and directories. Use "rm -r" to delete just the directory. It deletes both the folder and the files it contains when using only the rm command.
4. touch — The touch command is used to create a file. It can be anything, from an empty txt

file to an empty zip file. For example, “touch new.txt”.

1. man & --help — To know more about a command and how to use it, use the man command. It shows the manual pages of the command. For example, “man cd” shows the manual pages of the cd command. Typing in the command name and the argument helps it show which ways the command can be used (e.g., cd –help).
2. cp — Use the cp command to copy files through the command line. It takes two arguments: The first is the location of the file to be copied, the second is where to copy.
3. mv — Use the mv command to move files through the command line. We can also use the mv command to rename a file. For example, if we want to rename the file “text” to “new”, we can use “mv text new”. It takes the two arguments, just like the cp command.
4. locate — The locate command is used to locate a file in a Linux system, just like the search command in Windows. This command is useful when you don't know where a file is saved or the actual name of the file

# Chapter-4

## Version Control System

### Version Control System -Introduction

Version control systems are a class of software tools that help a software team manage changes to source code over the period of time. Version control software keeps track of ever y modification to the code in a special kind of database. If a mistake is made, developer s can turn back the clock and compare earlier versions of the code to help fix the mistakes while minimizing disruption to all team members.

Benefits of version control systems: Developing software without using version control is risky, like not having backups. Version control can also enable developer s to move faster and it allows software teams to preserve efficiency and agility as the team scales to include more developers. The primary benefit s you should expect from version control are as follows.

* + - A complete long-term change history of every file.
    - Branching and merging.
    - Traceability. Version Control Systems (VCS) have seen great improvements over the past few decades and some are better than others. VCS are sometimes known as SCM (Source Code Management) tools or RCS (Revision Control System)

### Git

* Introduction

Git is a mature, actively maintained open-source project originally developed in 2005 by Linus Torvalds, the famous creator of the Linux operating system kernel. A staggering number of software projects rely on Git for version control, including commercial projects as well as open-source. Developers who have worked with Git are well represented in the pool of available software development talent and it works well on a wide range of operating systems and IDEs (Integrated Development Environments).

Git is the best choice for most software teams today. While every team is different and should do their own analysis, here are the main reasons why version control with Git is preferred over alternatives.

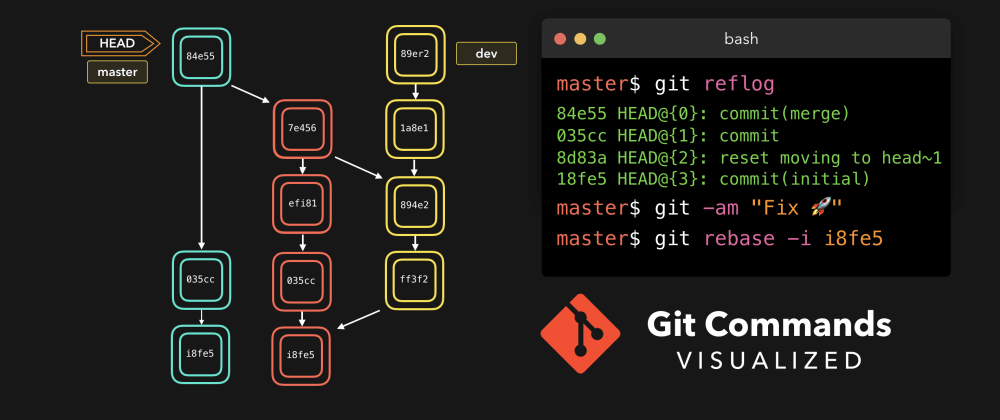
* Git has the functionality, performance, security and flexibility that most teams and individual

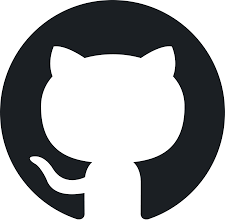
developers need.

* Git is the most broadly adopted tool of its kind.
* Git is a very well supported Test open-source project with over a decade of solid stewardship.
* Git enjoys great community support and a vast user base.
* Git has been designed with the integrity of managed source code as a top priority.
* Git is not fooled by the names of the files when determining what the storage and version history of the file tree should be, instead, Git focuses on the file content itself.
* It keeps track of changes to the code.
* Revert back to old version of the code.
* Test changes to code without losing the original.
* It synchronizes code between different people

#### Useful Commands

* git add
* git commit
* git status
* git push
* git pull
* merge conflict
* git log
* git reset
* branching
* git merge
* pull request



# Chapter-5

## IDE

### 5.1 Introduction to Project Setup

* + IDE: An integrated development environment (IDE) is software for building applications that combines common developer tools into a single graphical user interface (GUI). An IDE typically consists of:
  + Source code editor: A text editor that can assist in writing software code with features such as syntax highlighting with visual cues, providing language specific autocompletion, and checking for bugs as code is being written.
  + Local build automation: Utilities that automate simple, repeatable tasks as part of creating a local build of the software for use by the developer, like compiling computer source code into binary code, packaging binary code, and running automated tests.
  + Debugger: A program for testing other programs that can graphically display the location of a bug in the original code.

**Why do developers use IDEs?**

An IDE allows developers to start programming new applications quickly because multiple

utilities don’t need to be manually configured and integrated as part of the setup process. Developers also don’t need to spend hours individually learning how to use different tools when every utility is represented in the same workbench. This can be especially useful for onboarding new developers who can rely on an IDE to get up to speed on a team’s standard tools and workflows. In fact, most features of IDEs are meant to save time, like intelligent code completion and automated code generation, which removes the need to type out full character sequences.

* + SDK:

SDK stands for software development kit or devkit for short. It’s a set of software tools and programs used by developers to create applications for specific platforms. SDK tools will include a range of things, including libraries, documentation, code samples, processes, and guides that developers can use and integrate into their own apps. SDKs are designed to be used for specific platforms or programming languages.

Thus, you would need an Android SDK toolkit to build an Android app, an iOS SDK to build an iOS app, a VMware SDK for integrating with the VMware platform, or a Nordic SDK for building Bluetooth or wireless products, and so on.

The Characteristics of a Good SDK:

Because your mobile SDK is meant to be used outside your organization, it has to provide value to other businesses and their developers. That value is dependent on your SDK having the following characteristics:

* + Easy to use by other developers
  + Thorough documentation to explain how your code works
  + Enough functionality so it adds value to other apps
  + Does not negatively impact a mobile device’s CPU, battery, or data consumption
  + Plays well with other SDKs

In short, it just has to work. Ideally, it should work elegantly, but when time is of the essence, as long as it gets the job done, it should be good enough.

I have used IntelliJ IDEA and Pycharm to, I have also had the exposure of writing the spark queries and run the EMR Clusters to get the required data.

# Chapter-6

## System Requirement Analysis

* 1. **Information Gathering**

In Automation of measurement metrics we needed to identify, automate the generation of schedule sheets which can later on used on production jobs.

To get started with this I needed to Identify the Templates and which templates we are going to use in which format, Once I did analysis on this gather complete knowledge on it. I moved on to identify which library and languages should I proceed with.

With template finalization it gave me a more clear thought on libraries and languages that is supposed to be used. I end up up having two suggestions.

## System feasibility

For Automation of metrics generation I came up with two approaches in terms of Tech Stack.

### Not-Economical

**Python-Pandas:**

I came up with this approach but the problem was that pandas is used to handle large data, here, in our use case we don’t need that much amount of data so discarded because it is inefficient to our use-case.

### Economical

**Java with Either JExcel or Apache-Poi.**

I came up with another approach which is to use Java with libraries JExcel or Apache-Poi, so with this I finally concluded the Apache-Poi is best solution for it because it is lighter and handles all kind template files.

### Platform Specification

**6.3.2 Software Implementation Technology**

We used multiple softwares which includes Java, Gradle, Apache-Poi, and some another set of packages dependencies which were written in Java and use of Amazon S3

# Chapter-7

**In-Depth Technological Analysis**

* 1. **Storage**
     1. **Amazon S3**

Amazon Simple Storage Service (Amazon S3) is an object storage service offering industry- leading scalability, data availability, security, and performance. Customers of all sizes and industries can store and protect any amount of data for virtually any use case, such as data lakes, cloud-native applications, and mobile apps.

We have used this Storage service to store our schedule sheet generation templates and files, so that it will be handy whenever we need to access some information from it. We just need to to the bucket name and the key, and with these information we can access the required file from S3.

## Dynamo DB

Amazon DynamoDB is a fully managed NoSQL database service that provides fast and predictable performance with seamless scalability. DynamoDB lets you offload the administrative burdens of operating and scaling a distributed database so that you don't have to worry about hardware provisioning, setup and configuration, replication, software patching, or cluster scaling.

## Frame-Works

### Apache-Poi

Apache POI tutorial provides basic and advanced concepts of Apache POI technology. Our Apache POI tutorial is designed for beginners and professionals.

Apache POI is a Java library that is used to handle Microsoft Office Documents. Apache POI is open source, can be used by JVM based programming languages.

### Pandas

Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series.

* 1. **Languages**
     1. **JAVA**

Java platform is a collection of programs that help to develop and run programs written in the Java programming language. Java platform includes an execution engine, a compiler, and a set of libraries. JAVA is platform-independent language. It is not specific to any processor or operating system. Java is a programming language created by James Gosling from Sun Microsystems (Sun) in 1991.

The target of Java is to write a program once and then run this program on multiple operating systems. The first publicly available version of Java (Java 1.0) was released in 1995. Sun Microsystems was acquired by the Oracle Corporation in 2010. Oracle has now the steer Manship for Java. Oracle continues this project called OpenJDK.

Exposure in Java:

Used Optional data type which allows handles that no empty value is present.

Used Enum in Java, we although consider it a basic Enum but, overall it’s a powerful data type

where we can define our constants and and can access any values in Enum.

Used Immutatble maps: it helps us to create a map in which we can put those values which are not going to change.

Lambda functions and Streams in Java.

* + 1. **Python**

Python is a high-level, interpreted, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation.[30]

Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.

# Chapter-8

**Testing**

### Unit-Testing

Unit testing is a type of software testing where individual units or components of a software are tested. The purpose is to validate that each unit of the software code performs as expected. Unit testing is done during the development (coding phase) of an application by the developers.

For this I have used **Mockito-Junit and previously used unittest in python.**

### Integration-Testing

Integration Testing is a level of software testing where individual units / components are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. It helps us to analyze whether code that we have written is working according to the behavior that we have thought of.

I have also tried for a workaround to integration testing, i.e. to invoke the Api, with the given set of input parameters and trigger the workflow.

# Chapter-9

**Summary and Conclusion**

As my training is still going with Amazon, but so far in journey of 15 weeks, I was able to learn many things and was able to implement it simultaneously as well.

One of the things that I learnt in my internship is that how to automate the generation of metrics and how important and it is to choose appropriate tech-stack for individual component of a project, Initially I have coded it in Python only to realize that we are not dealing with the larger set of data, later on switched to Java Apache - Poi.

And worked on another Tasks which is to use Dao layers and implement it to interact with DynamoDB, in this part of task I realized the importance of abstraction, and just show the user what all they can use, and not the internal working of that Api.

This internship helped me to grow as a professional and I have learnt to work in a corporate culture under strict deadlines and supervision. I have also learnt to handle multiple projects at a time and work with a team. This is a full time internship where I have worked 60 hours a week.

I have also written test cases for the application under development that will help the development team to keep these test cases in mind and will help reduce time and energy in testing after the development is complete.

This training will also help in my professional career as a software developer while I would be able to look at the software under development with the insight of a developer and create better projects.

I have also learnt to follow strict deadlines and professionalism to be followed in meetings and corporate hierarchy.

Overall this project has been an excellent learning opportunity for me and will help me grow as an individual as well as a professional.

# Chapter-10

**Result and Discussion**

The proposed work for Automation of metrics generation will help to achieve the goal of developers to save a lot of time and hence providing efficiency to the production level jobs. This will help to reduce the manual task and manual spinning of the cluster to fetch the data manually and wasting hours of time in it.

This is not just a simple automation task, but it’s a complete solution that it can used help the developers save their time, save the industry time and money, so that they can scale up their task without any problems.

Another Tasks, that is to create Dao layers to interact with DynamoDB, was important task that had helped me to understand the importance of abstraction, that is write a DAO interface in such a manner that you are just exposing that what kind of inputs are needed and rest you do not have to think about implementation part of it.