

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

BELAGAVI-590 018, KARNATAKA



REPORT ON

“Inter/Intra Institutional Internship”

(21INT49)

Submitted in the partial fulfilment of requirement for the award of Degree

Submitted by

WASIA SAJID

USN : 4BD21CS183

Faculty Mentor

Prof. Anusha N

Department of CS&E

BIET, Davanagere



Department of Computer Science and Engineering

Bapuji Institute of Engineering and Technology

Davanagere-577004

2022-23

Bapuji Institute of Engineering and Technology

Davanagere-577004



Department of Computer Science and Engineering

CERTIFICATE

This is to certify that **WASIA SAJID** bearing USN **4BD21CS183**, of Computer Science and Engineering Department have satisfactorily submitted the report entitled “**Inter/Intra Institutional Internship**” (**21INT49**). The report has been approved as it satisfies the academic requirements for the year 2022-23.

INTERNSHIP COORDINATORS

Prof. Anusha N
Assistant Professor
Department of CS&E

Prof. Gangadharappa S
Assistant Professor
Department of CS&E

Dr. Naresh Patel KM
Assistant Professor
Department of CS&E

Head of Department

Dr. Nirmala C R
Professor & Head
Department of CS&E

Examiner 1:

Examiner 2:

DECLARATION

I, **WASIA SAJID**, USN : **4BD21CS183** , student of B.E in Computer Science and Engineering, **Bapuji Institute of Engineering and Technology, Davanagere**, hereby declare that the internship work entitled **“Inter/Intra Institutional Internship” (21INT49)** submitted to the **Visvesvaraya Technological University** during the academic year 2022-2023 is record of an original work done by us under the guidance of **Prof. Anusha N**, Department of Computer Science and Engineering, Bapuji Institute of Engineering and Technology Davanagere . This internship work is submitted in partial fulfillment of the requirements for the award of the degree of **Bachelor of Engineering in Computer Science and Engineering**. The results embodied in this report have not been submitted to any other university or institute for any degree.

WASIA SAJID

USN: 4BD21CS183

ACKNOWLEDGEMENT

Salutations to our beloved and highly esteemed institute, “**BAPUJI INSTITUTE OF ENGINEERING AND TECHNOLOGY**” for having well qualified staff and lab furnished with necessary equipment’s.

We express our sincere thanks to the **Genesis Company** for the opportunity to intern with your organization and learn valuable soft skills and basics of Java.

We express our sincere thanks to our faculty mentor **Prof. Anusha N** for giving us constant encouragement, support and valuable guidance throughout the course of project without whose guidance this project would not have been achieved.

We express our sincere thanks to our Internship Coordinators Prof. **Gangadharappa S and Dr. Naresh Patel K M** for giving us constant encouragement, support and valuable guidance throughout the course of project without whose guidance this project would not have been achieved.

We express whole hearted gratitude to **Dr. Nirmala C R**, H.O.D of Computer Science and Engineering Department. We wish to thank her for making our task easy by providing her valuable help and encouragement.

We also express our whole hearted gratitude to our principal, **Dr. H B Aravind** for his moral support and encouragement.

We would like to extend our gratitude to all the staff of **Computer Science and Engineering Department** for their help and support. We have benefited a lot from the feedback, suggestions given by them.

We would like to extend our gratitude to all our family members and friends for their advice and moral support.

COMPANY CERTIFICATE

GENESIS TRAINING


INTERNSHIP COMPLETION CERTIFICATE


This certificate is issued to Mr./Ms.

WASIA SAJID

With USN - 4BD21CS183 Studying in **COMPUTER SCIENCE & ENGINEERING**
at Bapuji Institute of Engineering & Technology, Davanagere
for the completion of a three-week internship from
October 12,2022 to November 04,2022

During this period, he/she has completed the internship on "Soft Skills,
Fundamentals of Java programming and Fundamentals of Android App
Development"


AKSHAY V AGASTHYA
VICE PRESIDENT (I&D)
GENESIS BENGALURU


NAVEEN NAGARAJ
MANAGING DIRECTOR
GENESIS BENGALURU

SUMMARY

The internship at Genesis was a comprehensive learning experience that encompassed both technical and non-technical aspects. In the realm of technical skills, I gained proficiency in Java programming and Android app development, equipping me with valuable tools for future software development endeavors. Additionally, I absorbed essential software development practices, troubleshooting techniques, and the importance of technical documentation.

Complementing these technical skills, my internship provided a platform for personal growth. I honed problem-solving abilities, adaptability, and resourcefulness when faced with challenges. Effective communication, teamwork, and time management skills were cultivated through real-world project work. Moreover, leadership development activities unveiled my leadership potential, while networking opportunities expanded my professional connections.

In essence, the Genesis internship was a transformative journey that not only elevated my technical competencies but also nurtured essential soft skills. This comprehensive skill set positions me confidently for a successful career in the dynamic technology industry, and I am deeply grateful to the Genesis team for their invaluable mentorship and support throughout this transformative experience. This internship has not only advanced my skill set but has also set a strong foundation for future career aspirations.

CONTENTS

Chapter	Title	Page no
Chapter 1	About the company	1
	1.1 Profile	1
	1.2 Services Provided	2
	1.3 Genesis Methodology	3
	1.4 Success Story	4
	1.5 Technology Services	5
	1.5.1 Software Development Life Cycle	5
	1.5.2 Scrum	8
	1.5.3 Azure	9
Chapter 2	Tasks performed	11
	2.1 Soft skills development	11
	2.2 Basics of java	13
	2.3 Android development	18
Chapter 3	Technical and Non Technical Activities	21
	3.1 Non Technical Activities	21
	3.2 Technical Activities	22
Chapter 4	Outcomes of Internship	23
	4.1 Learning from Internship Problem	23
	4.2 Technical Outcomes	23
	4.3 Non Technical Outcomes	24
Chapter 5	Conclusion	25
	Reference	26

LIST OF FIGURES

Figure no	Title of Figures	Page no.
5.1	Text	18
5.2	Button	18
5.3	Edit text	19
5.4	Image	19
5.5	Popup	20
5.6	Application	20

CHAPTER 1

ABOUT THE COMPANY

Genplus (Formerly know as Genesis Training) is an organization of keen, enthusiastic and driven individuals who are willing to push the boundaries of formal education and to explore and create practical and implementable strategies of learning and retention. Genesis was founded in 2015 by Naveen Nagaraj who is the Alumnus of BMS College of Engineering where Mr. Theertharaj was appointed as a CEO, who is the Alumnus of SJB Institute of Technology and Texas A&M University, USA.

Genesis is an innovative start-up in the field of Campus Recruitment Training. Their motto is to 'Create Values'. Keeping this in mind, they not only provide training and assess the students but also analyze the results, scientifically identify areas of improvement and provide strategic follow-up programs for the same. They believe that this, one of a kind, training program will better help the students to maximize their areas of strength and work on areas which require improvement.

Vision

Their vision is to create a platform where learning is no longer a burden but a choice made by that specific learner to transform their dreams into reality.

Mission

Their mission is to enhance the level of excellence amongst student and pave the way for a better career opportunity.

1.1 Profile

Genesis is built on values such as innovation, transparency and co-existence. Knowledge transfer, follow-up and data analysis are their key differentiator and their unique selling point. Their data analysis not only highlights the participant's or student's key strengths but also on their areas of improvement. They believe that this representation of the data enables the learner to better plan their study schedule.

1.2 Services Provided

1) Campus Recruitment Training

Refreshing & refining the skill sets, vital for an aspiring apprentice, to crack the initial evaluations of companies visiting their campus for the purpose of recruitment.

2) Technical Workshops

Simulated learning **emphasizing** on engaging the partakers to involve in collaborative and individual projects that are designed to enhance level of technical familiarity.

3) Placement Drive Management

Collaborative initiatives with the campus in driving company specific programs from Training to Placement process.

5) Corporate Training

Conducting need-based workshops to up skill the existing workforce and equip them with state-of-the-art dexterity to execute their everyday responsibilities.

6) Intellectual Training and Personality Development

Deliberations on the vastness of the power of knowing and embracing the understanding of the same into daily life to achieve a better self.

7) Staffing Solutions

Delivering qualitative personnel to the job market for varied sectors. Creating varied employment negotiations based on needs of companies associated with.

8) Software Research and Development

Designing client specific software, which are implementable and would enhance the efficiency in delivering its intent. Providing upgrades to existing systems to bring in intensity to performance.

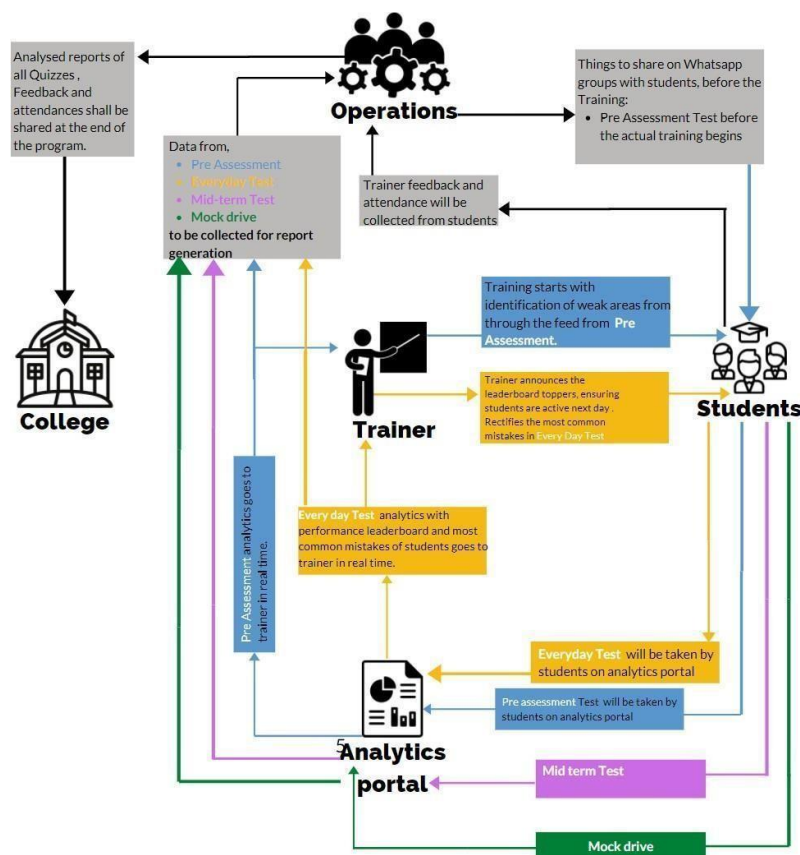
9) Collaborations

Creating platforms to visualize, ideate and bring in a combined effort to turn the same into reality. We appreciate the courage to dream and the will to build it and would like to associate ourselves in such splendid events.

1.3 Genesis Methodology

For their training program they adapt to one of a kind, tested and trusted model called **A-A-D-D**. This model helps them to deliver a result driven training program. A-A-D-D is a continues cycle which consists of 4 key elements:

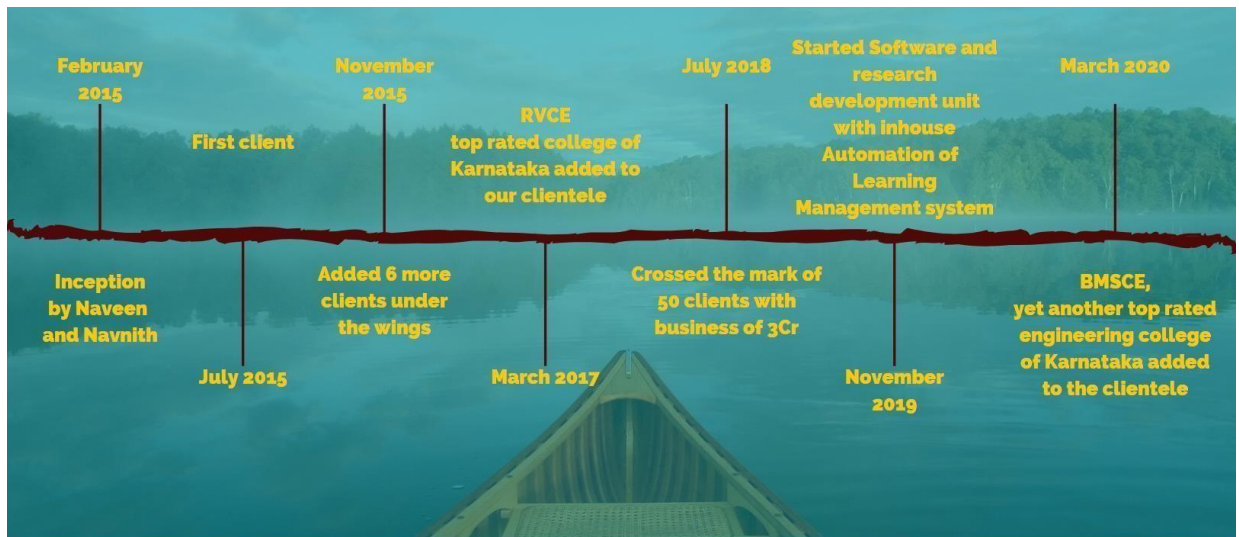
- **Assess:** Their first step into the program starts by assessing the students with a BASELINE assessment. This assessment is a blend of questions and patterns of all major recruiters.
- **Analyze:** The data from the assessment is crucially analyzed to identify the strengths and weaknesses of the students in various topics and subtopics.
- **Design:** They later design the module and plan of action to delivery specifically to the need.
- **Deliver:** This step focuses on executing the standard operating procedure designed in the previous step by our experienced and capable facilitators.



1.3 Genesis Methodology

1.4 Success Story

Founded in 2015, within a short span of 5 years we have 50+ clients and the clientele include Karnataka's top engineering colleges like RVCE and BMSCE. We could achieve this only because of our dedication to deliver results-oriented training. Genesis Training is an innovative start-up in the field of Campus Recruitment Training. Our motto is to 'Create Values'. Keeping this in mind, we not only provide training and assess the students, but we also analyze the results, scientifically identify areas of improvement and provide strategic follow-up programs for the same. We believe that this, one of a kind, training program will better help the students to maximize their areas of strength and work on areas which require improvement. We begin where our competitors end. We help transform ordinary to extraordinary!!!



1.4

Success Map

LEADERSHIP TEAM And Background

Genesis is founded by Naveen Nagaraj who is the Alumnus of BMS College of Engineering. Under his leadership Company had seen a tremendous success in a span of 5 years. To create an impact and accelerate in the EduTech industry, The company was registered as a **Genplus training and consulting services pvt ltd**. With the vision of creating a platform where learning is no longer a burden but a choice made by that specific learner to transform their dreams into reality. With this huge step of reorganization, The board decided to bring in a fresh young talent

to drive the corporate culture into the team, Where Mr. Theertharaj was appointed as a CEO, who is the Alumnus of SJB Institute of Technology and Texas A&M University, USA.

Theertharaj- CEO and BOD

Theertharaj, CEO, Genesis comes with 5+ years of industry experience from world leading cloud computing organization – Microsoft Corporation working at its Headquarters, Washington, USA. Holds a master's degree in Computer Engineering from Texas A&M University, USA. Having a wide experience in C, C++, C#, .Net, Microsoft Azure Technologies implementing highly scalable, reliable software solutions. As a passionate trainer, had trained for 3+ years on various technologies and Campus recruitment oriented technical curriculum.

Varun Gowda- VP Business development and BOD

The travel bug bit this person real hard and he decided to make a career out of it. A motivated and an experienced individual handling the Business expansion and sustenance. He bring with him a bankable experience in business operations and is currently the Vice President of Business development at Genesis

Akshay- VP L&D and BOD

A corporate trainer with 15 years of experience is an expert in Soft Skills and verbal ability training. Akshay has Bachelor's degree in commerce and also holds MA honours in English literature. He comes from an extremely rich corporate training background.

Naveen- Founder, BOD and Strategic advisor

A mechanical engineer who dreamed beyond his might and made it into reality. The founder of Genesis as an idea and company. He is a musician and keen on creating visual delights on screen. He is now the Board of Director of the company and assists in management level decisions

Chethan Simha- Technical Content and delivery advisor

Techie by passion, Alumnus of **RNSIT** and **University of Illinois, Chicago** who exactly knows what the industry needs. A strong believer of the statement "One who knows how to do can get a job anywhere but the one who knows why to do will be the other persons boss"

1.5 Technology Services

The companies' services on technology are listed below.

1.5.1 Software Development Life Cycle

SDLC or the Software Development Life Cycle is a process that produces software with the highest quality and lowest cost in the shortest time possible. SDLC provides a

well-structured flow of phases that help an organization to quickly produce high-quality software which is well tested and ready for production use.

The SDLC involves six phases as explained in the introduction. Popular SDLC models include the waterfall model, spiral model, and Agile model.

SDLC works by lowering the cost of software development while simultaneously improving quality and shortening production time. SDLC achieves these apparently divergent goals by following a plan that removes the typical pitfalls of software development projects. That plan starts by evaluating existing systems for deficiencies.

Next, it defines the requirements of the new system. It then creates the software through the stages of analysis, planning, design, development, testing, and deployment. By anticipating costly mistakes like failing to ask the end-user or client for feedback, SDLC can eliminate redundant rework and after-the-fact fixes.

It's also important to know that there is a strong focus on the testing phase. As the SDLC is a repetitive methodology, you have to ensure code quality at every cycle. Many organizations tend to spend few efforts on testing while a stronger focus on testing can save them a lot of rework, time, and money. Be smart and write the right types of tests.

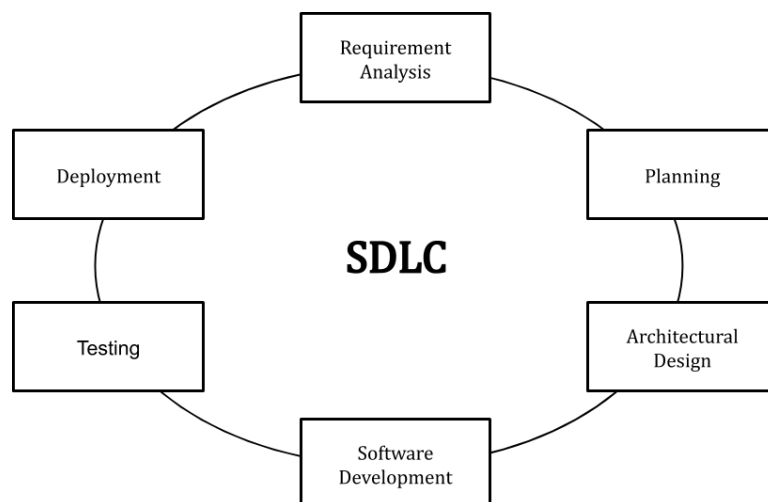


Fig 1.5.SDLC Life Cycle

Stages and Best Practices

Following are the best practices and stages of SDLC ensures the process works in a smooth, efficient, and productive way.

1. Identify the Current Problems

“What are the current problems?” This stage of the SDLC means getting input from all stakeholders, including customers, salespeople, industry experts, and programmers. Learn the strengths and weaknesses of the current system with improvement as the goal.

2. Plan

“What do we want?” In this stage of the SDLC, the team determines the cost and resources required for implementing the analyzed requirements. It also details the risks involved and provides sub-plans for softening those risks.

In other words, the team should determine the feasibility of the project and how they can implement the project successfully with the lowest risk in mind.

3. Design

“How will we get what we want?” This phase of the SDLC starts by turning the software specifications into a design plan called the Design Specification. All stakeholders then review this plan and offer feedback and suggestions. It’s crucial to have a plan for collecting and incorporating stakeholder input into this document. Failure at this stage will almost certainly result in cost overruns at best and the total collapse of the project at worst.

4. Build

“Let’s create what we want.” At this stage, the actual development starts. It’s important that every developer sticks to the agreed blueprint. Also, make sure you have proper guidelines in place about the code style and practices.

For example, define a nomenclature for files or define a variable naming style such as camelCase. This will help your team to produce organized and consistent code that is easier to understand but also to test during the next phase.

5. Code Test

“Did we get what we want?” In this stage, we test for defects and deficiencies. We fix those issues until the product meets the original specifications. In short, we want to verify if the code meets the defined requirements.

6. Software Deployment

“Let’s start using what we got.” At this stage, the goal is to deploy the software to the production environment so users can start using the product. However, many organizations choose to move the product through different deployment environments such as a testing or staging environment.

This allows any stakeholders to safely play with the product before releasing it to the market.

Besides, this allows any final mistakes to be caught before releasing the product.

1.5.2 Scrum

Scrum is a framework that allows development teams flexibility to respond to changing situations. This framework has sufficient control points in place to ensure the team does not stray from the desired outcome, and that issues can be identified and resolved and process adjustments made while the effort is still underway.

The Scrum Lifecycle starts with a prioritized backlog but does not provide any guidance as to how that backlog is developed or prioritized.

The Scrum Lifecycle consists of a series of Sprints, where the end result is a potentially shippable product increment. Inside of these sprints, all of the activities necessary for the development of the product occur on a small subset of the overall product. Below is a description of the key steps in the Scrum Lifecycle:

1. Establish the Product Backlog.
2. The product owner and development team conduct Sprint Planning. Determine the scope of the Sprint in the first part of Sprint Planning and the plan for delivering that scope in the second half of Sprint Planning.
3. As the Sprint progresses, development team perform the work necessary to deliver the selected product backlog items.
4. On a daily basis, the development team coordinate their work in a Daily Scrum.
5. At the end of the Sprint the development team delivers the Product Backlog Items selected during Sprint Planning. The development team holds a Sprint Review to show the customer the increment and get feedback. The development team and product owner also reflect on how the Sprint has proceeded so far and adapting their processes accordingly during a retrospective.
6. The Team repeats steps 2–5 until the desired outcome of the product have been met.

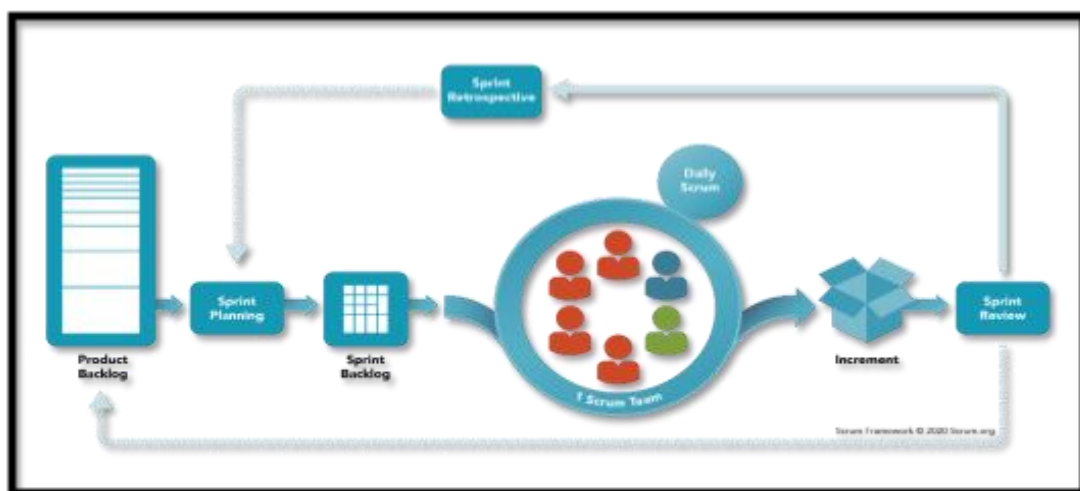


Fig1.5.2 Scrum Lifecycle

1.5.3 Azure

Microsoft Azure, formerly known as Windows Azure, is Microsoft's public cloud computing platform. It provides a range of cloud services, including compute, analytics, storage and networking. Users can pick and choose from these services to develop and scale new applications, or run existing applications in the public cloud.

The Azure platform aims to help businesses manage challenges and meet their organizational goals. It offers tools that support all industries -- including e-commerce, finance and a variety of Fortune 500 companies -- and is compatible with open source

technologies. This provides users with the flexibility to use their preferred tools and technologies. In addition, Azure offers 4 different forms of cloud computing: infrastructure as a service (IaaS), platform as a service (PaaS), software as a service (SaaS) and serverless.

Microsoft charges for Azure on a pay-as-you-go basis, meaning subscribers receive a bill each month that only charges them for the specific resources they have used.

Working of Azure

Once customers subscribe to Azure, they have access to all the services included in the Azure portal. Subscribers can use these services to create cloud-based resources, such as virtual machines (VM) and databases.

Because Microsoft Azure consists of numerous service offerings, its use cases are extremely diverse. Running virtual machines or containers in the cloud is one of the most popular uses for Microsoft Azure. These compute resources can host infrastructure components, such as domain name system (DNS) servers; Windows Server services -- such as Internet Information Services (IIS); or third-party applications. Microsoft also supports the use of third-party operating systems, such as Linux. Azure is also commonly used as a platform for hosting databases in the cloud.

CHAPTER 2

Tasks performed

2.1 Soft skill development

1. Definition of Soft Skills:

Soft skills encompass a wide range of personal qualities and attributes. They include interpersonal skills, communication skills, emotional intelligence, and other non-technical skills that enable individuals to work effectively with others, handle social situations, and navigate their personal and professional lives successfully.

2. Versatility:

Unlike hard skills, which are job-specific and often technical in nature, soft skills are versatile. They can be applied in various contexts, industries, and roles. Whether you're a manager, a customer service representative, or an artist, soft skills play a vital role in your ability to interact, collaborate, and succeed.

3. Complement to Hard Skills:

Soft skills complement hard skills, creating a well-rounded and effective individual. While hard skills are essential for performing specific tasks, soft skills help individuals communicate, adapt, solve problems, and work harmoniously with others to achieve goals.

4. Communication:

Effective communication involves conveying ideas, information, and emotions clearly and concisely. It encompasses both verbal and non-verbal communication, active listening, and the ability to articulate thoughts persuasively. Strong communication skills are fundamental in personal and professional relationships.

5. Emotional Intelligence:

Emotional intelligence (EQ) is the ability to recognize, understand, manage, and use emotions effectively. It includes self-awareness (understanding one's own emotions), empathy (recognizing and understanding others' emotions), and interpersonal skills (managing relationships and interactions with others).

6. Teamwork:

Teamwork involves collaborating with others to achieve common objectives. Soft skills related to teamwork include effective communication within a team, cooperation, compromise, and the ability to contribute positively to group dynamics. Successful teamwork is vital in most workplaces.

7. Problem Solving:

Problem-solving skills enable individuals to analyze situations, identify challenges or obstacles, and find effective solutions. Critical thinking, creativity, and the ability to make informed decisions are essential components of problem-solving.

8. Adaptability:

Adaptability is the capacity to adjust to changing circumstances and embrace new ideas or methods. It includes being flexible, resilient, and open to innovation. In today's rapidly evolving world, adaptability is a valuable skill.

9. Leadership:

Leadership involves guiding and motivating individuals or teams to achieve common goals. Leadership soft skills include communication, decision-making, delegation, and the ability to inspire and influence others. Leadership is not limited to formal roles; it can be exhibited at all levels of an organization.

10. Time Management:

Time management skills encompass the ability to organize tasks, set priorities, and meet deadlines efficiently. Effective time management ensures that individuals use their time wisely, reduce stress, and achieve their goals in both personal and professional life.

2.2 Basics of java

2.2.1 Platform Independence:

Java is a platform-independent language, which means that Java code can run on any platform (e.g., Windows, macOS, Linux) without modification, as long as there is a compatible Java Virtual Machine (JVM) installed.

Write Once, Run Anywhere (WORA): Java's platform independence is often summarized as "Write Once, Run Anywhere." This means that you can write your Java code on one platform (e.g., Windows) and run it on any other platform (e.g., macOS or Linux) without modification, as long as there's a compatible Java Virtual Machine (JVM) installed on that platform.

Bytecode Compilation: Java source code is first compiled into an intermediate form known as bytecode. This bytecode is not specific to any particular platform and is highly portable.

Execution by JVM: The JVM acts as a runtime environment for Java applications. It interprets and executes the bytecode, making the code platform-independent. Each platform has its own JVM implementation, which allows Java to run on various operating systems.

2.2.2 Object-Oriented:

Java is an object-oriented programming (OOP) language. It uses classes and objects to structure code and promote reusability. Everything in Java is an object, except for primitive data types.

Objects and Classes: In Java, everything is based on objects and classes. An object is an instance of a class, and a class is a blueprint or template for creating objects. Classes define the properties (attributes) and behaviors (methods) that objects of that class will have.

Encapsulation: Encapsulation is a key principle of OOP, and Java enforces it. It means bundling data (attributes) and methods (functions) that operate on that data into a single unit, i.e., a class. This hides the implementation details from the outside and allows for data access control.

Inheritance: Java supports inheritance, which is the ability of a class (subclass or child class) to inherit properties and behaviors from another class (superclass or parent class). Inheritance promotes code reuse and the creation of class hierarchies.

Polymorphism: Polymorphism means the ability of objects of different classes to be treated as objects of a common superclass. In Java, this is often achieved through method overriding, where a subclass provides a specific implementation of a method declared in its superclass.

Abstraction: Abstraction involves simplifying complex reality by modeling classes based on the essential properties and behaviors. In Java, you can use abstract classes and interfaces to achieve abstraction.

Modularity: OOP encourages modularity, which means breaking down a complex system into smaller, manageable components (classes). Java's class-based structure naturally promotes modularity, making it easier to develop, maintain, and understand code.

Encapsulation: Encapsulation involves hiding the internal details of an object and providing a well-defined interface for interacting with it. In Java, you can use access modifiers like `private`, `protected`, and `public` to control access to class members (fields and methods).

Reusability: One of the main benefits of OOP is code reusability. You can create reusable classes and libraries that can be used in multiple projects, reducing development time and effort.

Data Abstraction: Java allows you to represent real-world entities as objects, abstracting away the complexity of data structures and algorithms. This simplifies problem-solving and modeling.

Real-World Modeling: OOP encourages developers to model software systems after real-world entities and relationships, making code more intuitive and easier to maintain.

2.2.3 Main Method:

Every Java program starts with a `public static void main(String[] args)` method. This is the entry point of a Java application.

Entry Point: In Java, the main method serves as the entry point for a program. When you execute a Java application, the Java Virtual Machine (JVM) looks for the main method to start execution.

Access Modifiers: The main method is declared with the access modifiers `public`, `static`, and `void`. Here's what these mean:

- **public:** This modifier indicates that the main method is accessible from outside the class.
- **static:** The main method is associated with the class itself rather than with instances of the class. This allows it to be called without creating an object of the class.
- **void:** `void` specifies that the main method doesn't return any value.

Method Name: The name of the method is always `main`. This is not case-sensitive, so you can use `Main` or `MAIN`, but by convention, it's written in lowercase as `main`.

Parameters: The main method accepts an array of strings as a parameter, denoted as `String[] args`. These are known as command-line arguments. You can use this array to pass information to your Java program when it's executed.

Execution: When you run a Java program, the JVM looks for the main method and starts executing code from there. Any statements within the main method's block `{ ... }` will be executed in sequence.

In the example above, MyJavaProgram is the name of the Java class, and the main method is where the program starts. It prints "Hello, World!" to the console when executed.

This main method is essential in Java programs, as it provides a starting point for the execution of your code. It's the first method that the JVM looks for and calls when you run your Java application.

2.2.4 Data Types:

Java has both primitive data types (e.g., int, float, boolean) and reference data types (e.g., objects, arrays). Primitive types are not objects, while reference types are.

(i) Primitive Data Types:

Integers:

- byte: 8-bit signed integer.
- short: 16-bit signed integer.
- int: 32-bit signed integer (most commonly used for integers).
- long: 64-bit signed integer (useful for large integers).

Floating-Point Numbers:

- float: 32-bit floating-point number.
- double: 64-bit floating-point number (commonly used for floating-point calculations).

Characters:

- char: 16-bit Unicode character (used for storing single characters).

Boolean:

- boolean: Represents a true or false value (used for conditional logic).

(ii) Reference Data Types:

- **Objects:** Objects in Java are instances of classes. They are reference data types because they don't directly contain the data but rather a reference (memory address) to where the data is stored.
- **Arrays:** Arrays in Java are also reference data types. They can hold multiple values of the same data type. Array elements are accessed via indexing.
- **User-Defined Types:** Java allows you to create your own reference data types by defining classes. These classes can have attributes (fields) and methods.

2.2.5 Variables in Java:

- Variables are used to store data, such as numbers, text, or objects, in Java programs.
- Before using a variable, you must declare it by specifying its name and data type.

- The data type determines the kind of data the variable can hold (e.g., int for integers, double for floating-point numbers, String for text).
- Properly declaring and initializing variables is a fundamental practice in Java programming, ensuring data is stored and used correctly in your code.

2.2.6 Control Flow:

Java provides control flow statements like if, else, while, for, switch, and do-while for decision-making and looping.

Decision-Making:

Java offers conditional statements like if, else, and else if for making decisions in your code. These statements allow you to execute specific blocks of code based on conditions.

Looping:

Loops in Java, including while, for, and do-while, enable repetitive execution of code. Loops are used when you want to perform a task multiple times, such as iterating through arrays or processing data.

Switch Statement:

The switch statement provides an efficient way to select one of many code blocks to be executed based on the value of a variable or expression. It's often used for multiple-choice scenarios.

Control Flow Structures:

These control flow structures allow you to control the flow of your program, ensuring that specific parts of your code execute under certain conditions. For example, you can use if to execute code when a condition is true, or while to repeat code as long as a condition is true.

Branching:

Control flow statements help your program branch into different paths depending on conditions. For example, you can use if and else to provide alternate paths of execution.

Iteration:

Loops like for and while are used for iteration, where a block of code is executed repeatedly until a specific condition is met.

Termination Conditions:

In looping constructs like while and for, you must define termination conditions to avoid infinite loops. An infinite loop can cause your program to hang or crash.

Nested Control Flow:

Java allows nesting of control flow statements. For example, you can place an if statement within a while loop to create complex decision-making structures.

2.2.7 Packages:

Java uses packages to organize classes into namespaces. Packages help in avoiding naming conflicts and organizing code.

Organizing Code: Packages are used in Java to organize classes and interfaces into logical groups or namespaces. This helps in managing and structuring large codebases.

Avoiding Naming Conflicts: Packages prevent naming conflicts between classes or interfaces with the same name. By placing related classes in a package, you can use the package name as a prefix to distinguish them.

Access Control: Packages also control access to classes and their members. Java provides access modifiers like public, protected, private, and package-private (default) to specify the level of visibility for classes and members within a package.

Java's Package Hierarchy: Java has a hierarchical package structure. Standard Java libraries (Java API) are organized into packages, such as java.util, java.io, java.lang, etc. Your code can also be organized into packages to mimic this structure.

Package Declaration: To declare a class or interface as part of a package, you use the package keyword at the beginning of the source code file. For example, package com.example.myapp; declares that the class belongs to the com.example.myapp package.

Import Statements: To use classes or interfaces from other packages, you can use import statements. For example, import java.util.ArrayList; allows you to use the ArrayList class from the java.util package without fully qualifying its name.

Creating Your Packages: You can create your custom packages to organize your code. The package structure in your file system should match the package declaration in your code. For example, if you declare package com.example.myapp;, your file should be in a directory structure like com/example/myapp.

Package Naming Conventions: Package names are typically written in all lowercase letters to adhere to Java naming conventions. They often use a reversed domain name to ensure uniqueness, like com.example.myapp.

Benefits of Packages: Packages enhance code maintainability, reusability, and collaboration among developers. They make it easier to locate and manage related classes and reduce the risk of naming conflicts.

2.3 android development

Text:

Displaying Text:

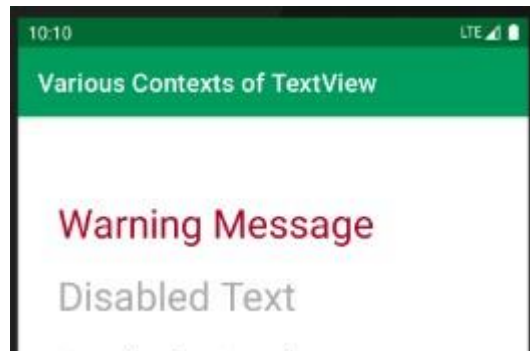
Use the TextView widget to display static text.
Set text programmatically using the setText() method.

Formatting Text:

Use HTML tags or spans for text formatting within TextView.

Localization:

Store strings in resource files (strings.xml) for easy translation.



5.1

(text)

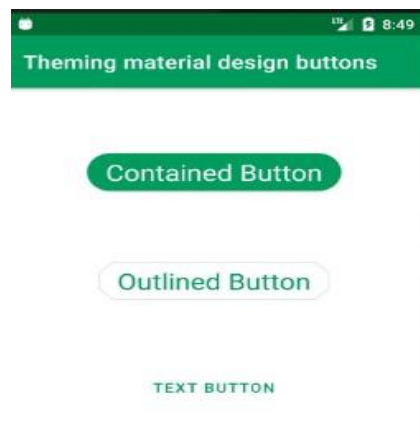
Button:

Creating a Button:

Use the Button widget to create clickable buttons.
Define an OnClickListener to handle button clicks.

Button Actions:

Perform actions like navigation, form submission, or triggering events in the OnClickListener.



5.2

(button)

Edit Text:

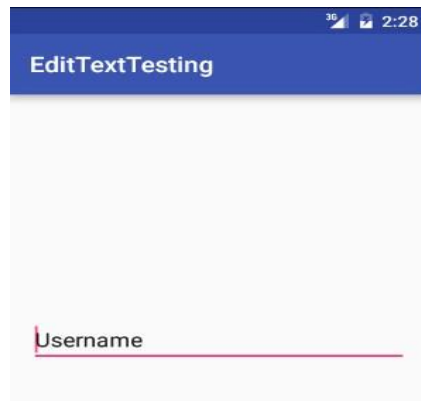
Adding Input Fields:

Use the EditText widget to create text input fields.
Retrieve user input using getText().toString().

Input Validation:

Validate user input for correctness (e.g., email, password).

Show error messages for invalid input.



5.3

(input text)

Image:**Displaying Images:**

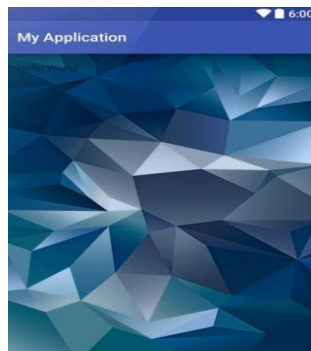
Use the ImageView widget to display images.

Load images from resources or URLs using libraries like Picasso or Glide.

Image Scaling:

Configure scaling options such as center cropping or fit-center.

Adjust image dimensions using XML attributes or programmatically.



5.4(image)

Popup:**Creating Popups (Dialogs):**

Use Dialog or AlertDialog for creating popups.

Customize dialog content and appearance.

Dialog Actions:

Implement actions like OK, Cancel, or custom actions using dialog buttons.

Handle button clicks within the dialog.

5.5



(popup)

Formula (Java):

Implementing Formulas:

Use Java classes and methods to implement formulas.

Example: Calculate the result of a mathematical formula.

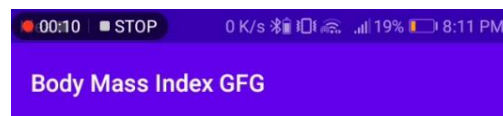
Input Handling:

Accept user input as parameters for formula calculations.

Validate input before performing calculations.

Displaying Results:

Show the result of formula calculations in a TextView or other UI elements.



Body Mass Index Calculator

Weight(in Kg)

70

Height (in cm)

176

The BMI is

22.59814

Healthy

RECALCULATE

5.6 (application)

Chapter 3

Technical and Non Technical Activities

3.1 Non-Technical Activities (Soft Skill Development):

During my internship at Genesis, I had the opportunity to engage in various non-technical activities aimed at enhancing my soft skills. These activities were instrumental in improving my interpersonal, communication, and leadership skills. Some of the key non-technical activities I participated in include:

Communication Skills Workshops: Attendance in workshops and training sessions focused on improving communication skills, including effective written and verbal communication.

Team Building Exercises: Participation in team-building activities and projects to learn the importance of collaboration and effective teamwork.

Leadership Development: Active involvement in leadership development programs that allowed me to take on leadership roles within project teams.

Time Management Training: Acquisition of valuable time management techniques for efficient organization of work and meeting deadlines.

Networking Opportunities: Genesis provided ample opportunities for networking with professionals in the field, expanding my professional network and exposing me to industry insights.

3.2 Technical Activities (Basics of Java and Android Development):

In addition to the non-technical activities, I had the privilege of diving into the realm of technical skills, specifically learning the fundamentals of Java programming and Android app development. Here are the technical activities I undertook during my internship:

Java Programming: Comprehensive training in Java programming, which included learning syntax, data structures, and object-oriented programming concepts. Application of this knowledge to develop small-scale Java applications.

Syntax: Understanding the syntax and structure of Java code, which includes variables, data types, loops, conditionals, and functions.

Object-Oriented Programming (OOP): Gaining proficiency in the principles of object-oriented programming, such as classes, objects, inheritance, encapsulation, and polymorphism.

Data Structures: Exploring essential data structures like arrays, lists, and maps, which are crucial for organizing and manipulating data efficiently.

Error Handling: Learning how to handle exceptions and errors gracefully in Java programs.

Android Development: Engagement in understanding Android app development, including learning about Android Studio, user interface design, and creating basic Android applications. Successful completion of a project involving building a simple Android app.

Android Studio: Learning how to use Android Studio, the official Integrated Development Environment (IDE) for Android app development, to create and manage projects.

User Interface Design: Exploring the principles of designing user-friendly interfaces, including the use of XML layouts and widgets to create visually appealing and responsive app interfaces.

Activity Lifecycle: Understanding the lifecycle of Android activities and how to manage them effectively, including handling user interactions and device rotations.

Basic App Creation: Successfully completing a project that involved building a simple Android app. This project allowed me to apply my knowledge of Java and Android development principles to create a functional mobile application.

Throughout my internship, the guidance and mentorship provided by the experienced professionals at Genesis were invaluable in my technical skill development.

In conclusion, my internship at Genesis was a fulfilling experience that allowed me to enhance my soft skills while acquiring a solid foundation in Java and Android development. I am grateful for the opportunities and support I received during my time here and believe these experiences have equipped me with valuable skills for my future career endeavors.

Chapter 4

Outcomes of Internship

4.1 Learning from Internship Problems:

During my internship at Genesis, I encountered and addressed several challenges and problems. These experiences were invaluable in terms of learning opportunities and personal growth. Some key takeaways from the problems I encountered include:

Problem-Solving Skills: I developed strong problem-solving skills by tackling real-world challenges in the workplace. I learned how to analyze problems, break them down into manageable parts, and devise effective solutions.

Adaptability: I gained the ability to adapt to new and unexpected situations. Facing various issues taught me to be flexible and open to change, a crucial skill in the ever-evolving tech industry.

Resourcefulness: To solve problems, I often needed to research and seek out resources independently. This improved my research skills and resourcefulness in finding solutions.

Communication: Dealing with problems often required effective communication with team members and supervisors. I improved my ability to articulate issues and propose solutions clearly and concisely.

Learning from Failure: Not every problem had an immediate solution, and I learned the importance of learning from failure. Each setback provided an opportunity for growth and improvement.

4.2 Technical Outcomes:

My internship at Genesis had a significant impact on my technical skills and knowledge. The technical outcomes of my internship include:

Java Proficiency: I gained a solid understanding of Java programming, including syntax, object-oriented principles, and data structures. This knowledge enables me to write efficient and maintainable Java code.

Android Development Skills: I acquired fundamental skills in Android app development, allowing me to design and build basic Android applications. This skillset positions me well for further exploration of mobile app development.

Software Development Practices: I learned best practices for software development, including version control, code review processes, and project management methodologies. These practices are essential for producing high-quality software in a professional setting.

Debugging and Troubleshooting: I developed expertise in debugging and troubleshooting software issues, which is a critical skill for identifying and resolving problems in complex systems.

Technical Documentation: I improved my ability to create technical documentation, such as code comments and user manuals, which enhances collaboration and knowledge sharing within development teams.

4.3 Non-Technical Outcomes:

Beyond technical skills, my internship at Genesis also had non-technical outcomes that contributed to my personal and professional growth:

Communication Skills: I honed my communication skills through regular interactions with colleagues and supervisors. Effective communication is essential for collaboration and conveying ideas clearly.

Teamwork and Collaboration: Working in project teams exposed me to the dynamics of collaboration. I learned how to contribute to a team's success by leveraging individual strengths and supporting team goals.

Time Management: Meeting project deadlines and balancing multiple tasks improved my time management skills. Efficient time management is crucial for productivity and project success.

Leadership Potential: Participating in leadership development activities allowed me to recognize my leadership potential and gain confidence in taking on leadership roles in the future.

Networking: Engaging in networking opportunities expanded my professional network, providing connections and insights that may be valuable for my future career endeavors.

Chapter 5

Conclusion:

My time at Genesis as an intern was an enriching and transformative experience that offered a balanced mix of technical and non-technical learning opportunities. Throughout this internship, I encountered various challenges that, when overcome, not only honed my problem-solving skills but also taught me to adapt, be resourceful, and communicate effectively.

On the technical side of things, I made significant strides in my Java programming proficiency, enabling me to grasp fundamental concepts and principles better. Additionally, my exposure to Android development provided a solid foundation for mobile app development expertise. I gained insights into software development practices, debugging techniques, and the importance of thorough technical documentation – all of which are indispensable in the technology sector.

Beyond just technical skills, my time at Genesis also fostered personal growth. I refined my communication abilities, delved into the subtleties of effective teamwork, and improved my time management skills. Engaging in leadership development activities awakened my leadership potential, while networking opportunities expanded my professional connections.

In summary, my internship at Genesis wasn't merely a period of skill acquisition; it was a transformative journey of personal and professional development. The experiences, knowledge, and skills I acquired during this internship have equipped me with the confidence to navigate the dynamic and competitive landscape of the technology industry. I am deeply thankful to the entire Genesis team for their mentorship and unwavering support throughout this transformative experience.

This internship has proven to be a significant stepping stone toward achieving my career aspirations, and I eagerly anticipate applying the lessons learned and skills acquired in my future pursuits.

References

<https://ieeexplore.ieee.org/document/9465935>

<https://ieeexplore.ieee.org/document/7003608>

<https://ieeexplore.ieee.org/document/7503306>

<https://developer.android.com/>

<https://developer.android.com/studio>

<https://stackoverflow.com/questions/tagged/android>

<https://docs.oracle.com/en/java/>

<https://www.coursera.org/specializations/java-programming>

<https://www.edx.org/learn/soft-skills>