INTERNSHIP PROJECT REPORT

(Jan 2022-June 2022)

Internship report submitted in partial fulfilment of the requirement for the

degree of Bachelor of Technology

in

Computer Science and Engineering

By

Ayush Shukla

181227

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology,

Waknaghat, 173234,

Himachal Pradesh, INDIA

DECLARATION

I hereby declare that this submission is my own work carried out at Cognizant Technology Solutions Corporation, Noida from January 2022 to June 2022 and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma from a university or other institute of higher learning, except where due acknowledgment has been made in the text.

Submitted by:
Ayush Shukla
181227
Computer Science & Engineering and Information Technology Department
Jaypee University of Information Technology

ACKNOWLEDGEMENT

I would like to thank Ms. Supriya and all the people that worked along with me at Cognizant Technology Solutions Corporation, Noida with their patience and openness they created an enjoyable working environment.

I also would like to thank Mr. Rosary for mentoring me throughout my internship.

It is indeed with a great sense of pleasure and immense sense of gratitude that I acknowledge the help of these individuals.

I am highly indebted to Mr. Pankaj Kumar, Training & Placement Coordinator of our college for the facilities provided to accomplish this internship. I would also like to thank the Head of our Department Dr.Vivek Kumar Sehgal and the faculty for teaching us the skills required for this internship.

Finally, I must acknowledge the constant support and patients of my parents.

Ayush Shukla
181227
Jaypee University of Information Technology

TABLE OF CONTENT

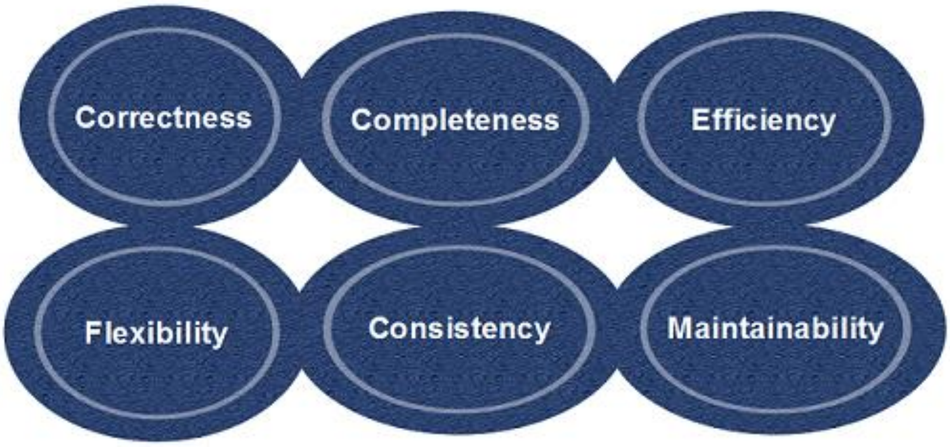
Chapter 01: INTRODUCTION

1.1 Introduction

An internship is a professional teachable moment that gives students real-world experience in their field of study or professional goals. An internship enables students to learn new skills while exploring and enhancing their careers. This report is a description of my ongoing internship at Cognizant Technology Solutions Corporation**.** This internship report details the activities that helped me achieve a bunch of my stated objectives. I was assigned the profile of ".NET and Full Stack Developer" for my internship. First 2 months was learning of the basic concepts of engineering and web development following the 3rd month will based on company project.

1.2 Job Description

The introduction of engineering concepts to the creation, implementation, and technical management is known as software engineering. Software engineering was designed to address the challenges of low-quality software initiatives. Problems arise when software exceeds schedules, budgets, and quality expectations. It assures that the software is built in a consistent, correct, timely, cost-effective, and specification-compliant manner. Software engineering became vital to keep up with the rapid depending of user needs and the context in which the program is expected to operate.



Objectives of Software Design

The Software Engineer Trainee is responsible to assist in the design and development of software. The Software Engineer Trainee works with other members of the team to develop secure and dependable software solutions. The duties and responsibilities of this position are as follows:

- Application development (coding, programming)
- Code debugging and testing
- New software application documentation and testing
- Investigating, diagnosing, and resolving a wide range of technical challenges
- Working with senior executives
- Identifying issues and developing solutions
- Learning about new technology ahead of time

1.3 Objectives

The objective of the internship is to learn more about the programming languages like Java, MySQL, JavaScript and ReactJS. These languages are the basic requirements for my job profile. After the learning process, I was assigned to work on multiple mini projects for the company. This internship provides experience to the freshers so that they can learn more about the industry. The perspective was to gain enough knowledge so that we can work easily on the project assigned to us.

1.4 Internship Schedule

The internship plan was as follows:

➢ January 2022

I was designated my role and domain as full stack software engineer in Java. Two weeks of orientation took places and several meetings and activities took place to accustom us to the company.

➢ February 2022

Learning phase began and was carried out in both supervised as well as self-paced methods. We started receiving training in basic core concepts of computer science. We had a refresher course that included a basic course on web development, database, object-oriented programming and data structures and algorithms.

➢ March 2022

We received training on advanced topics related to our domain like Advanced Java, Spring framework and tools and techniques related to debugging and good code writing practices.

➢ April 2022

We received in depth training on Spring tool suite which included technologies like Spring core, Maven, Spring boot, Java Persistence application programming interface and Spring REST.

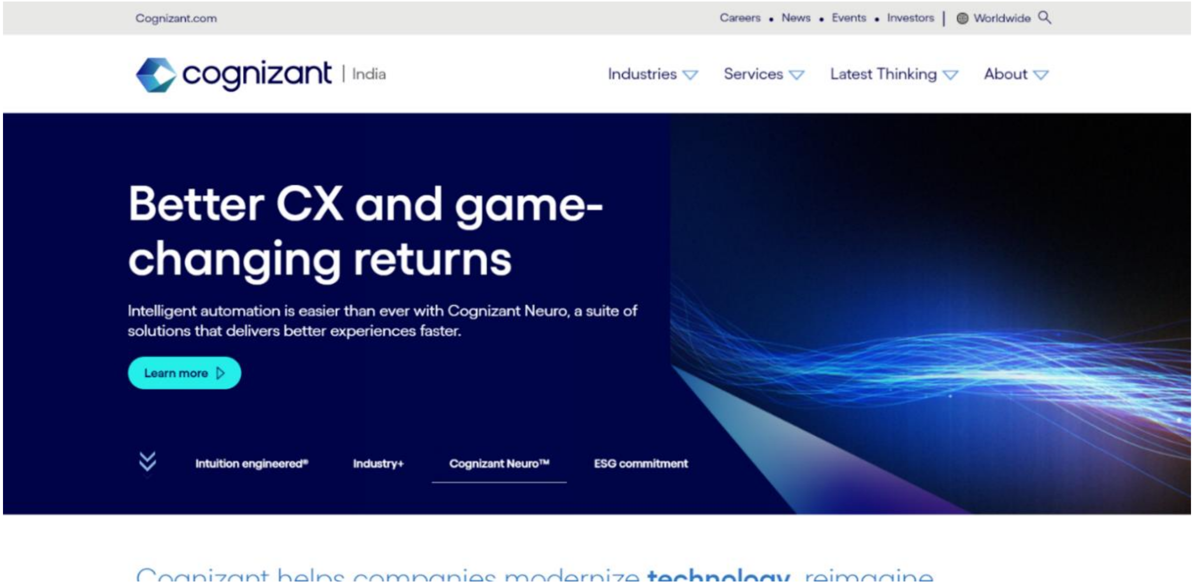
➢ May 2022

We were assigned to multiple mini projects and a major project that represented real life client-based requirements. Out of that one project of the most advance one has been mentioned in detail further in the report

Chapter 02: COMPANY DESCRIPTION

2.1 Cognizant

Cognizant is a multinational consulting and services firm located in Teaneck, New Jersey. Cognizant offers IT consulting and business services to a variety of industries, including banking, health care, manufacturing, media, and entertainment.



Better CX and game-changing returns

Intelligent automation is easier than ever with Cognizant Neuro, a suite of solutions that delivers better experiences faster.

Cognizant helps companies modernize **technology**, reimagine

Fig 2. About Cognizant

2.1.1 Our Providers

Enabling the Cloud - Cognizant is there to help you no matter where you are on your cloud transition journey. We work with you to develop the optimal strategy for your company so that you may reap the full benefits and value of cloud computing. Our vision, industry expertise, and client-centric approach enable us to provide end-to-end cloud services and unique solutions that lead to short-term success, measurable business outcomes, and satisfied customers. Learn how Cognizant can assist you in accelerating your cloud journey.

Application modernization services include the transfer of legacy applications or platforms to new applications or platforms, as well as the integration of new functionality to provide the most up-to-date features to the business.

AI analytics is a subset of business intelligence that uses machine learning techniques to discover new patterns and relationships in data. In practise, AI analytics is the process of automating much of the work that a data analyst would normally do.

2.2 Cognizant's History

Dun & Bradstreet Satyam Software (DBSS), with Srini Raju as the initial CEO and MD, was established as Dun & Bradstreet's in-house technology arm, focusing on executing large IT projects for Dun & Bradstreet corporations. In 1996, the firm began pursuing customers outside of Dun & Bradstreet.

Cognizant Corporation, situated in Chennai, India, was formed in 1996 when Dun & Bradstreet split off some of its companies, including Erisco, IMS International, Nielsen Media Research, Pilot Software, Strategic Technologies, and DBSS. After a few months, in 1997, DBSS changed its name to Cognizant Software Solutions. In July 1997, Dun & Bradstreet paid $3.4 million for Satyam's 24 percent stake in DBSS. In March 1998, the company's headquarters were relocated to the United States, and Kumar Mahadeva was named CEO. As part of Cognizant Corporation, the company focuses on Y2K-related work and web development.

In 1998, the parent company, Cognizant Corporation, split into two companies: IMS Health and Nielsen Media Research. Following the restructure, Cognizant Technology Solutions became a public division of

IMS Health. In June 1998, IMS Health separated the firm in half, making a public offering of Cognizant shares. The company can raise $34 million, which is less than IMS Health's underwriters had hoped for. They put money aside for debt payments and office improvements.

Kumar Mahadeva chose to reduce the company's reliance on Y2K contracts, with Y2K projects accounting for only 26% of revenues in the first quarter of 1999, compared to 49% in early 1998. Because he believed the $16.6 billion corporate software market was saturated, Kumar Mahadeva avoided large-scale ERP deployment projects. He instead focused on application management, which accounted for 37% of Cognizant's sales in the first quarter of 1999. Cognizant's revenues in 2002 were $229 million, and it had no debt and $100 million in total. The company flourished during the dotcom bust by taking on maintenance work that major IT services companies didn't want.

In 2003, IMS Health purchased the whole 56 percent stake in Cognizant, and the business instituted a poison pill strategy to prevent hostile takeovers. Kumar Mahadeva stepped down as CEO in 2003, and Lakshmi Narayanan took his place. The company gradually expanded its product portfolio beyond IT to include outsourced (BPO) and consulting services. In 2006, Francisco D'Souza succeeded Lakshmi Narayanan. From 2003 to 2012, Cognizant made ten consecutive placements on Fortune magazine's "100 Fastest-Growing Companies" list, demonstrating its tremendous growth.

In September 2014, Cognizant announced its largest acquisition, paying $2.7 billion for healthcare IT services company TriZetto Corp. Cognizant's stock increased by nearly 3% in pre-market trading.

On June 24, 2015, the company announced a multimillion-dollar agreement with Escorts Group in India to help the company's businesses with digital transformation and modernisation across all business areas.

On June 30, 2015, it partnered with NTUC FairPrice, a Singapore-based market chain, to digitally modernise NTUC's operations in order to provide a more personalised and consistent customer experience across several channels.

In April 2018, Cognizant and a group of Indian life insurers announced they were developing a blockchain solution to boost efficiency by permitting cross-company data sharing. The platform, which is built on R3's Corda distributed ledger technology platform, is believed to eliminate the need for third-party data brokers and aggregators for KYC, clinical and financial assessment, risk assessment, fraud detection, and regulatory compliance. Cognizant sold its Oy Samlink acquisition to Kyndryl in January 2022.

Chapter 03: TOOLS AND TECHNOLOGIES

3.1 ReactJS

ReactJS is a JavaScript framework for creating user interface components that are declarative, quick, and adaptive. It's a component-based front-end framework that's solely responsible for the app's view layer. Jordan Walke, a Facebook software engineer, came up with the concept. It was created and is maintained by Facebook, and it is later utilised in Facebook products like as WhatsApp and Instagram. ReactJS was created in 2011 for Facebook's feed area, although it was first widely released in May 2013.

MVC is the most used website architecture today (model view controller). In an MVC design, React is the view, while Flux or Redux is the architecture. A ReactJS application is made up of several components, each of which is responsible for producing reusable HTML code. All React programmes are built on the foundation of components. Simple building blocks can be used to create complex applications by nesting them with additional components. The DOM-based technique for populating data in the HTML DOM is interconnected by ReactJS. It is quick because the virtual DOM modifies individual DOM components rather than reloading the entire DOM.

We develop React components that correlate to many aspects to make a React app. These components, which are grouped into higher-level components, make up the application structure. Consider a form that has, among other things, the needed fields, labels, and buttons. Each form element can be built as a React component., which we then combine to build the form component. The form elements would describe the structure and parts of the form.

The main goal of ReactJS is to create a user interface that assists programme performance. It speeds up the programme by using a virtual DOM (JavaScript objects). The virtual DOM is faster than the regular DOM in JavaScript. ReactJS can be used on both the server and client sides, as well as in conjunction with other frameworks. To frame and make app maintenance easier, component and data structures are employed.

3.1.1 Features of ReactJS

ReactJS is gradually gaining traction among web developers as the most popular JavaScript framework. It's an essential component of the front-end ecosystem. Some of ReactJS' important features are as follows:

- JSX – JavaScript XML
- Components
- One-way Data Binding
- Virtual DOM
- Simplicity
- Performance

3.2 GitHub

Software developers and engineers can use GitHub to build public-facing cloud repositories for free. You can transfer a GitHub repository to your device, add and alter files locally, and then "push" your changes back to the repository, where they will be visible to the public.

Repository

A repository (sometimes shortened as "repo") is a site where all of the files for a project are kept. Each project has its repository, which you can access using a specific URL.

Forking a Repo

When you establish a new project based on an existing project, this is known as "forking." This is a fantastic tool that greatly fosters the development of new applications and projects. If you find a project on GitHub that you'd want to contribute to, you can fork it, make the modifications you want, and then republish it as a new repo. You may quickly add changes to your current fork if the original repository that you forked to create your new project is updated.

Git Commit

You're ready to commit once you've staged the files you want to add. The commit message is important regardless of whether you commit using GitHub Desktop or the command line. Short commit statements that describe your change should be used. The commit messages will lead you through the history of your repository, therefore they should be informative. The following message format can be used in command-line commits:

```
git commit -m "git commit message example"
```

Pull Requests

You've forked a repository, made a fantastic change to the project, and want the original devs to notice it—perhaps even put it in the original project/repository. Create a pull request to accomplish this. The original repository's authors can view your work and decide whether or not to accept it into the official project. When you submit a pull request, GitHub provides an excellent communication channel between you and the primary project's maintainer.

3.3 Core Java & Advanced Java

It is a Java programming language feature. It's a Java technology or version that's specifically built for developing web-based, network-centric, or enterprise applications. It covers topics such as Servlet, JSP, JDBC, RMI, and socket programming, among others. It is a domain-specific specialty.

The majority of advanced Java programmes employ a two-tier architecture, namely Client and Server. Advanced Java apps include all applications that execute on the server.

Why bother with Java?

- It reduces the complexity of developing a multi-tier application.
- Creates a common API for components and the application server container.
- The framework services are provided by the JEE application server and containers.
- Advantages of Advanced Java
- Advance Java has four primary advantages: network centricity, process simplification, and a futuristic imaging standard.
- JEE (advanced Java) includes libraries that help you comprehend the Client-Server architecture for web-based applications.
- Web and application servers such as Apache Tomcat and Glassfish are also available. We can learn how the HTTP protocol works by using these servers. It is not possible in core Java.
- It's also crucial to know advanced Java if you're working with trading technologies like Hadoop, cloud-native, and data science
- It provides a set of services, APIs, and protocols that enable the capability required to construct multi-tiered and web-based applications.

- A number of advanced Java frameworks, such as Spring, Hibernate, and Struts, enable us to construct secure transaction-based web applications like banking and inventory management.

Difference between Core Java and Advance Java

Criteria	Core Java	Advance Java
Used for	It is used to develop general purpose application.	It is used to develop web-based applications.
Purpose	It does not deal with database, socket programming, etc.	It deals with socket programming, DOM, and networking applications.
Architecture	It is a single tier architecture.	It is a mute-tier architecture.
Edition	It is a Java Standard Edition.	It is a Java Enterprise Edition.
Package	It provides java.lang.* package.	It provides java.servlet.* package.

3.4 Spring Tool Suite

All of these issues have been addressed by the Spring framework. This framework develops corporate applications utilising modern approaches such as Aspect-Oriented Programming (AOP), Plain Old Java Object (POJO), and dependency injection (DI), reducing the complexity associated with developing enterprise apps using EJB. Spring is a lightweight open source framework that enables Java EE 7 developers to create easy, dependable, and scalable corporate applications. This framework is primarily concerned with offering multiple methods for managing business items. When compared to traditional Java frameworks and Application Programming Interfaces (APIs), such as Java database connection (JDBC), Java Server Pages (JSP), and Java Servlet, it makes developing Web applications considerably easier. The Spring framework, which includes Spring AOP, can be thought of as a collection of sub-frameworks, often known as layers. Object-Relational Mapping in Spring (Spring ORM). Spring Web MVC, Spring Web Flow When building a Web application, you can use any of these modules separately. To improve the functionality of a Web application, the modules can be joined together.

Features of the Spring Framework

The Spring framework stands out from the crowd thanks to features like IoC, AOP, and transaction management. The following are some of the most important characteristics of the Spring framework:

- **IoC container:**

  The DI or IoC pattern is used by the core container to implicitly supply an object reference in a class during runtime. As an alternative to the service locator pattern, this pattern is used. The IoC container contains assembler code that controls application object configuration management.

The Spring framework includes two packages, org.springframework.beans and org.springframework.context, that help provide IoC container capabilities.

- **Data access framework:**

  Allows developers to store persistence data in databases using persistence APIs such as JDBC and Hibernate. It assists developers in handling a variety of issues, including how to communicate with a database connection, how to ensure that the connection is closed, how to handle exceptions, and how to perform transaction management. It also makes it simple for developers to implement code to access persistence data throughout the programme.

- **Spring MVC framework:**

  Allows you to create Web applications using the MVC framework. All user requests pass via the controller before being routed to various views, such as JSP pages or Servlets. All major view technologies, including as ISP, Jasper Report, FreeMarker, and Velocity, may simply interact with the Spring MVC framework's form handling and form validation functionalities.

- **Transaction management:**

  Helps an application handle transaction management without altering its code. This framework provides the Java Transaction API (JTA) for managing global transactions via an application server and local transactions using JDBC Hibernate, Java Data Objects (JDO), or other data access APIs. Spring's declarative and programmatic transaction management allows the developer to model a wide range of transactions.

- **Spring Web Service:**

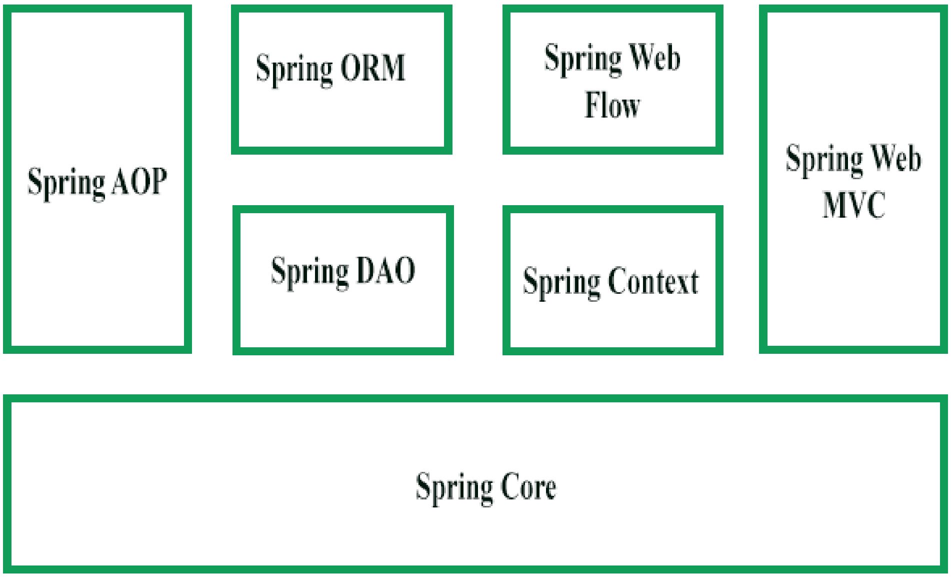
  Creates Web service endpoints and definitions from Java classes, however managing them in an application is tricky. Spring Web Service provides layered-based techniques to handle this challenge, each of which is controlled individually using XML parsing (the technique of reading and manipulating XML). Spring allows the developer to quickly distribute XML messages (objects) across two machines by providing effective mapping for sending incoming XML message requests to an object.

- **JDBC abstraction layer:**

  Aids users in dealing with faults in a simple and efficient manner. When this abstraction layer is used in a Web application, the JDBC programming code can be minimised. DriverNotFound exceptions are handled by this layer. The DataAccessException class is used to translate all SQLExceptions. Data Access Objects (DAO) are not bound to JDBC only because Spring's data access exception is not JDBC specific.

- **Spring Test Context framework:**

  Provides unit and integration testing capabilities for Spring applications. Furthermore, the Spring Test Context framework includes integration testing features like context management and test fixture caching DI, as well as transactional test management with default rollback semantics.



The Spring framework is made up of seven modules, as indicated in the diagram above. Spring Core, Spring AOP, Spring Web MVC, Spring DAO, Spring ORM, Spring context, and Spring Web flow are the modules in question. These modules provide several platforms for creating various enterprise applications; for example, the Spring Web MVC module can be used to create MVC-based apps.

Spring Framework Modules

- Spring Core Module:
- Spring AOP Module:
- Spring ORM Module:
- Spring Web MVC Module:
- Spring Web Flow Module:
- Spring Web DAO Module:
- Spring Application Context Module

3.5 Java Persistence API

The Java persistence API is abbreviated as JPA. The Java persistence API is a specification for persistence, which is a broad term that refers to any way by which Java objects can outlast their application activity. Although all Java objects do not need to be persistent, most applications nevertheless include critical business objects. Its definition allows you to specify which items in your Java applications must survive and how they should persist. It is not a tool or framework in and of itself; rather, it describes a set of principles that any tool or framework can enforce.

Although Hibernate was founded on JPA's object-relation mapping (ORM), it has since evolved. Similarly, while JPA was created for use with relational/SQL databases, it has been used with NoSQL datastores were extended. EclipseLink is a common platform that supports JPA with NoSQL and is the reference implementation of JPA 2.2.

What is JPA ORM?

Although they differ in execution, each JPA implementation includes some type of ORM layer. To comprehend JPA and JPA-compatible technologies, you must have a solid understanding of ORM. Object mapping is a technique that developers should avoid doing by hand. A framework like Hibernate ORM or EclipseLink converts this method into a library or application, or an ORM layer. The ORM layer, which is part of the framework architecture, performs the translation of software objects into a relational database.

The ORM layer transforms Java classes and objects into relational databases, allowing them to be saved and handled in Java. Fields are changed to columns, and the name of the object kept by design becomes the table name. Once the table is configured, the application focuses on each table row. Although object mapping can be changed, it works fine by default.

Data Persistence in Java

From a programming standpoint, the ORM layer is an interface layer. It translates Object Graph languages into SQL and relational tables. The ORM layer enables object-driven developers to design data collection apps without having to leave the object-driven paradigm. When you use the datastore, you establish a map for your application's data model items.

Rather than saving and retrieving objects, you create the mapping between your objects and your database and use JPA to persist them. If you're using a reference database, the Java Database Connectivity API JDBC can take care of a lot of the actual communication between your application code and the server. It adds annotations to metadata to define how things are mapped to databases in a specific form. For JPA annotations, each JPA implementation provides its own engine.

Configuring in JPA

It uses traditional coding, which, like other modern frames, provides a default setup based on industry best practises. A musician class, for example, would be mapped to the Musician database table. The default configuration is a time saver and works effectively in most cases. You can also customise your JPA setup.

Entity Relationships in JPA

The persistence of an object with a primitive field is only a half equation. It can also coordinate the management of institutions. For four different types of business relationships, tables and artefacts are available.

1. One to one
2. Many to one
3. One to many
4. Many to Many

The relationship between an entity and other entities is described by each form of relationship. A one-to-many relationship might exist between an entity represented by a collection like List or Set and Performance. If the musicians possessed a band-field, they may use a multi-to-one arrangement, implying that they were all in the same band class.

3.6 Maven

Maven is a project management and comprehension tool that provides a comprehensive build lifecycle framework for developers. Because Maven uses an uniform directory architecture and a default build lifecycle, the development team may automate the project's build infrastructure in nearly no time.

In the case of different development teams, Maven can quickly set up the way to operate according to standards. Maven makes life easier for developers when developing reports, inspections, build and testing automation setups because most project setups are simple and reusable.

Maven provides developers ways to manage the following −

- Builds
- Documentation
- Reporting
- Dependencies
- SCMs
- Releases
- Distribution
- Mailing list

To summary, Maven makes the project development process easier and more consistent. It effortlessly handles duties such as compilation, distribution, documentation, team collaboration, and others. Maven improves reusability and handles the majority of build-related chores.

Maven Development

Maven was created to help with the construction of the Jakarta Turbine project. There were a number of projects, each with somewhat different ANT build files. CVS was used to check in JARs.

After that, the Apache group created Maven, which can build many projects simultaneously, publish project information, deploy projects, exchange JARs across various projects, and aid team cooperation.

Objective

The primary goal of Maven is to provide developer with the following −

- A comprehensive model for projects, which is reusable, maintainable, and easier to comprehend.
- Plugins or tools that interact with this declarative model.

The structure and contents of a Maven project are declared in an xml file called pom.xml, which is the fundamental unit of the entire Maven system. We'll go through POM in more detail in coming chapters.

Convention trumps customization.

Developers are not required to establish their own build process because Maven leverages Convention over Configuration.

Developers are not required to include every single configuration detail. For projects, Maven provides sensible default behaviour. Maven creates a default project structure when you create a Maven project. The developer is simply required to place files in the appropriate locations and does not need to define any pom.xml configuration.

The default values for project source code files, resource files, and other configurations are shown in the table below. Assuming, ${basedir} denotes the project location −

Maven gives developers the ability to specify life-cycle goals and project requirements while building the project (that rely on Maven plugin capabilities and on its default conventions). Maven plugins handle a lot of the project management and build-related duties.

Developers do not need to understand how specific plugins function to build any Maven project. Maven Plugins will be covered in depth in later chapters.

Features of Maven

- A straightforward project setup that adheres to best practises.
- Consistent application throughout all projects.
- Management of dependencies, including automatic updates.
- A huge and increasing library collection.
- Extensible, providing the ability to develop Java or scripting language plugins quickly.
- New features are available right away with minimal or no configuration.
- Model-based builds Maven may generate predefined output types such as jar, war, and metadata from any number of projects.
- Project information site that is consistent Maven may generate a website and a PDF with complete documentation using the same metadata as the build process.
- Project release management and distribution publication Maven integrates with your source control system, such as CVS, and maintains project releases without any further configuration.
- Backward compatibility: You may easily transfer numerous project modules from prior Maven versions to Maven 3. It can also support older versions.
- Parent versioning is automatic, therefore there's no need to indicate the parent in the sub module for maintenance.
- It examines the project dependency tree and allows you to develop schedule modules in parallel. You can gain performance improvements of 20-50 percent with this method.
- Improved Error and Integrity Reporting Maven has improved error

reporting and now includes a link to the Maven wiki page where you can get a complete description of the error.

3.7 Microservices

A microservice is a distributed service that is tiny and loosely connected. Microservice architectures arose as a response to the scalability and innovation issues that Monolith designs posed (Monolith systems are often large, containing over 100,000 lines of code). It lets you to break down a large application into smaller, more manageable components with clearly defined roles.

Reasons for using Microservice:

There are a few issues in monolith application:

1. It can be difficult to understand the complexity of a huge programme and make code changes quickly and effectively; it can also be challenging to manage the code.

2. To ensure the impact of changes, applications must undergo comprehensive manual testing.

3. The entire application must be created and deployed for a little modification.

4. The large application slows down the start-up process.

Benefits of Microservices:

1. **Small Modules –**

   The application is divided down into smaller components that developers can easily build and manage.

2. **Easier Process Adaption –**

   Microservices make new technology and process adaptation easier. With the modern microservices that we utilise, you can experiment with new technologies.

3. **Independent scaling –**

   Based on their demands, each microservice can scale separately via X-axis scaling (cloning with extra CPU or memory) and Z-axis scaling (sharding).
4. **Unaffected –**

   The failure of a single module has little impact on large applications.
5. **DURS –**

   Each service can have its own DURS (deployed, updated, replaced, and scaled)..

Spring Boot is the best Java framework to implement Microservices.

3.8 AWS – Amazon Web Services

AWS is an Amazon service that leverages a distributed IT architecture to make various IT resources available on demand. Infrastructure as a service (IaaS), platform as a service (PaaS), and packaged software as a service are some of the services it offers (SaaS).

Amazon launched AWS, a cloud computing platform that allows businesses to benefit from dependable IT infrastructure.

Uses of AWS

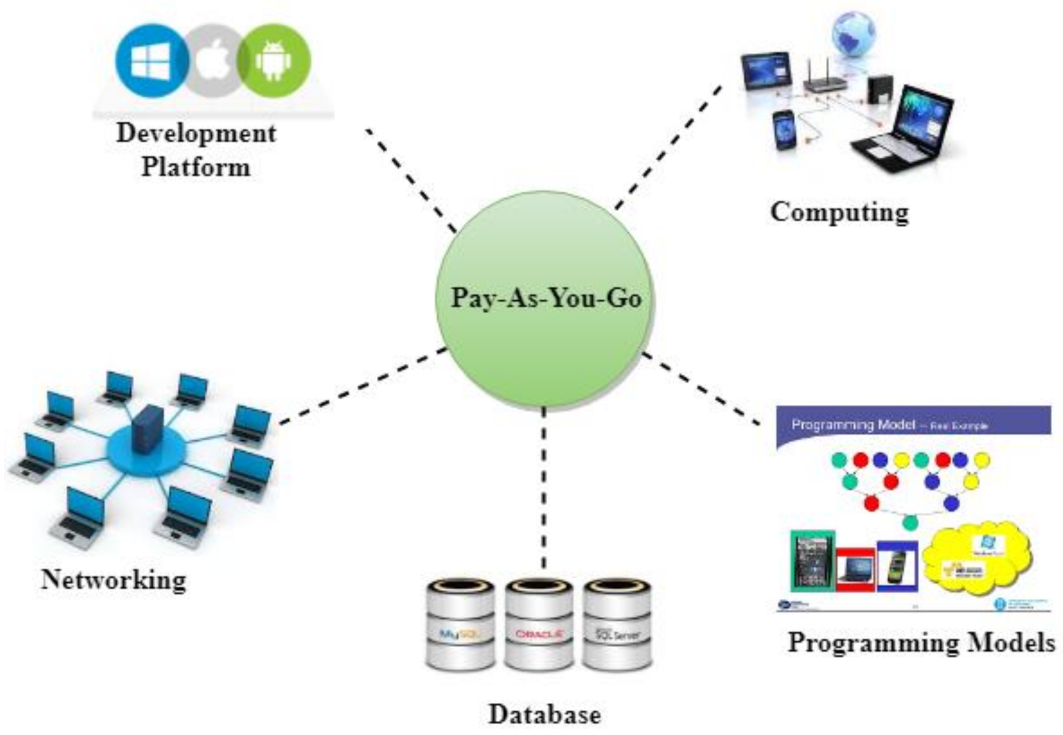
- A tiny manufacturing company leverages its expertise to develop its business by outsourcing IT administration to AWS.
- A multinational corporation with offices all around the world can use AWS to train its scattered workforce.
- AWS can be used by an architecture consulting firm to obtain a high-resolution representation of a construction prototype.
- A media organisation can utilise the AWS to distribute various

types of material to users all over the world, such as ebox or audio files.

Pay-As-You-Go

AWS provides services to consumers based on the Pay-As-You-Go philosophy. Customers can use AWS services whenever they choose, with no commitment or upfront cost. Customers can buy AWS services on a pay-as-you-go basis.

- Programming models
- Database storage
- Computing
- Networking



Advantages of AWS

1) Flexibility

2) Cost-effectiveness

3) Scalability/Elasticity

4) Security

Chapter 04: PROJECT

4.1 Description

A leading HealthCare Management Organization wanted to strengthen its Middleware by exposing the core logic related to Claims Management as Microservices. This middle ware Microservices was hosted on Cloud so that all the up/downstream applications can get an access to this for performing business transactions.

There was also a Member Portal developed as part of this scope that consumed these Microservices and responses back to members who are in need of Claim related information.

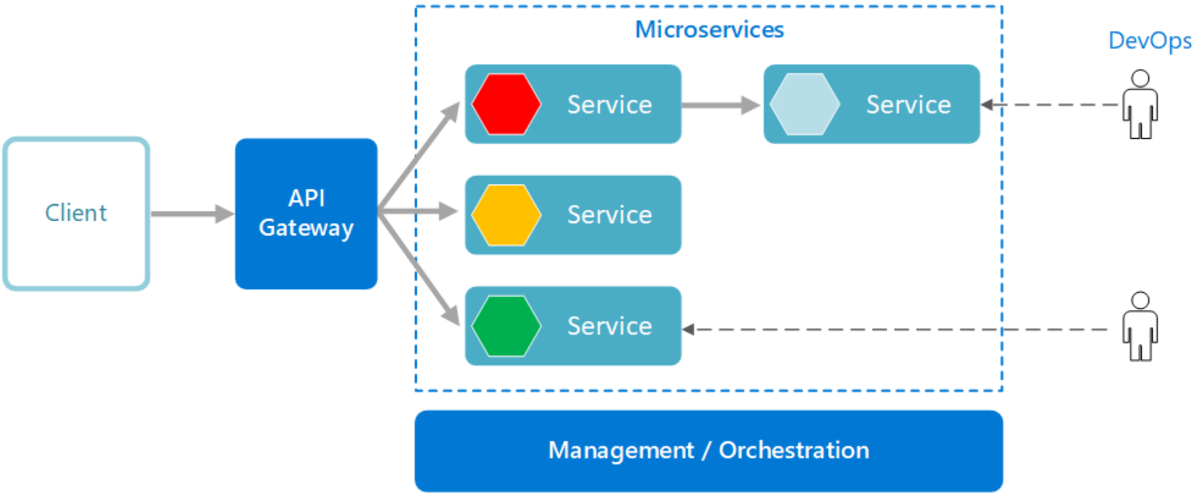
Below are the modules that were developed as part of the Project:

Req. No.	**Req. Name**	**Req. Description**
REQ_01	Claims Module	Claims Module is a Middleware Microservice that performs following operations: • Get Claim Status • Validate Eligibility of Claim and Action Settlement
REQ_02	Member Module	Member Module is a Middleware Microservice that performs the following operations: • View Bills • Submit Claim • View Claim Status
REQ_03	Policy Module	Policy Module is a Middleware Microservice that performs the following operations: • Get Chain of Permissible Providers (Hospitals) • Get Benefits permissible under a policy • Get Acceptable Claim Amount per benefit, per policy

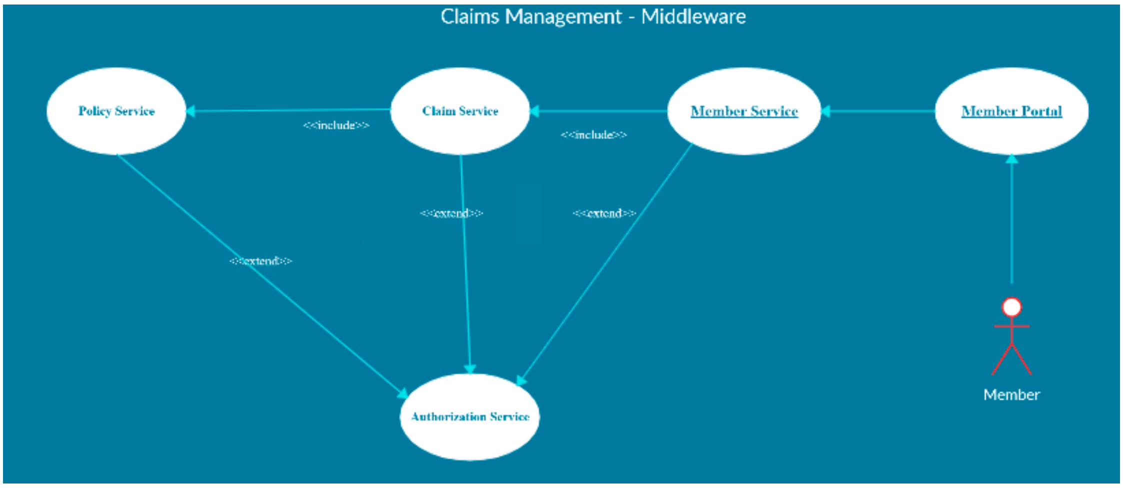
REQ_04	Member Portal	A Web Portal that allows a member to Login and allows to do following operations: • Login • View Current Bill Status, Next Due • Verify Claim Status • Submit a Claim

Hardware and Software Requirement

1. Hardware Requirement:
   a. Developer Desktop PC with 8GB RAM
2. Software Requirement (Java)
   a. Spring Tool Suite (STS) Or any Latest Eclipse
      i. Have PMD Plugin, EclEmma Code Coverage Plugin and AWS Code Commit Enabled
      ii. Configure Maven in Eclipse
   b. Maven
   c. Docker (Optional)
   d. Postman Client in Chrome



System Architecture Diagram



Use Case Diagram

Individual Components of the System

Member Microservice

<table>
<tr><td><b>Claims Management System</b></td><td><b>Member Microservice</b></td></tr>
<tr><td colspan="2">
<b>Functionalities</b>
Can assume that Member Portal App is the only client to this Microservice. An authorized member views the premium bills, submit the claim, can view the claim status, which is already submitted.

Post Authorization, basic Member based validation are performed in this Microservice and then it communicates to the below Microservices for retrieving necessary information.

Member Microservice will interact with the Claim Microservice for the following functionalities:
○ the Microservice will interact with Claims Module, to check the eligibility of the member and the claim, and then action the claim to set for processing.
○ To get the status of an already submitted claim
</td></tr>
</table>

Claims Microservice

Claims Management System	**Claims Microservice**
Functionalities Member Microservice interacts with Claims Microservice. Post authorization of request, Claim Microservice allows the following operations: To view the status of submitted claim: o Retrieve the claim status from database and return To verify claim eligibility by interacting with Policy Microservice and action settlement: o View the Claim details and check the following: i. If the Claimed Amount is applicable under the subscribed policy ii. If the Claimed benefit is applicable under the subscribed policy iii. If the Hospital in which benefits are availed is a permissible Health Care Provider (Hospital). o If the above 3 conditions are satisfied, update the claim as “Pending Action” else “Claim Rejected”. If any information is not available or found to be invalid, then update status as “Insufficient Claim Details”. If any contradictory details found, update status as “Under Dispute”.	

Policy Microservice

Claims Management System	**Policy Microservice**
Functionalities Claims Microservice interacts with Policy Microservice. Post authorization of request, Claim Microservice allows the following operations: o Provide the permissible providers in which healthcare services can be offered. (Return the chain of hospitals) o Provide the list of benefits which the member is eligible to, under a	

subscribed policy.

- To provide the eligible claim amount, for the given benefit under a subscribed policy.

Authorization Microservice

Claims Management System	**Authorization Microservice**
Security Requirements o Service to Service communication has to happen using JWT o Pass End User Context across Microservices o Have the token expired after specific amount of time say 15 minutes. o Have this service configured in the cloud along with other services	

Swagger

Claims Management System	**Swagger**
Documentation Requirements (Java) o All the Microservices must be configured with Swagger for documentation o Register the swagger resources in the Swagger Microservice and enable them as REST end points o Configure this service along with other services in the cloud	

Member Portal (MVC)

Claims Management System	**Member Portal**
Client Portal Requirements o Member Portal must allow a member to Login. Once successfully logged in, the member does the following operations:	

- View Claim Status
- Submit a Claim
- View Bill Status of the subscribed policies

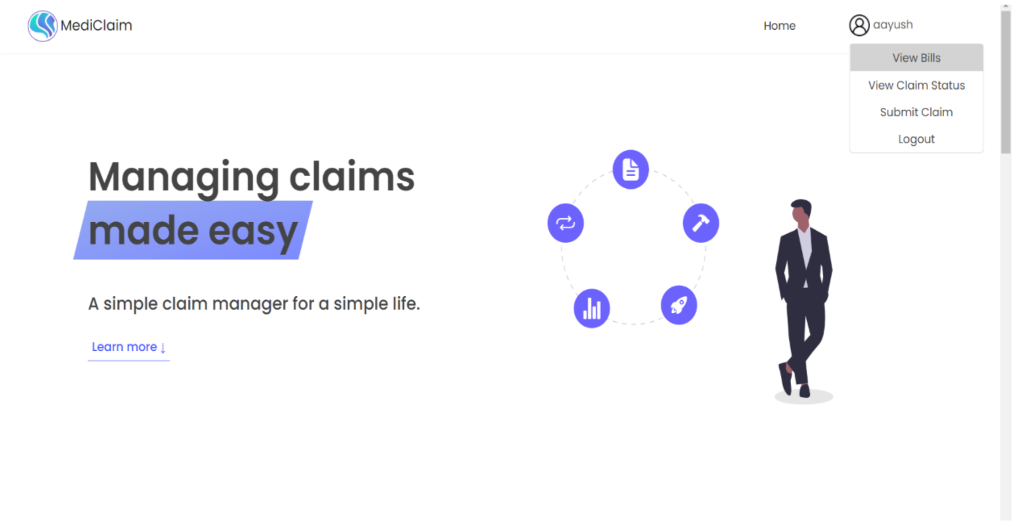
- Each of the above operations will reach out to the middleware Microservices that are hosted in cloud.

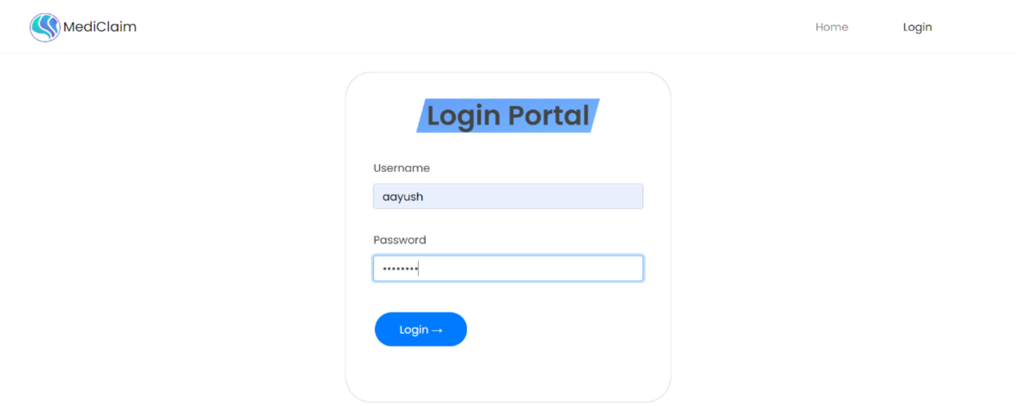
Cloud Deployment requirements

- All the Microservices must be deployed in Cloud
- All the Microservices must be independently deployable. They have to use In-memory database or user sessions wherever applicable
- The Microservices has to be dockized and these containers must be hosted in Cloud using CI/CD pipelines
- The containers have to be orchestrated using AWS/Azure Kubernetes Services.
- These services must be consumed from an MVC app running in a local environment.

4.5 Results

Here are some snapshots, depicting various modules and functionalities of the project and how the front end looks and operates.

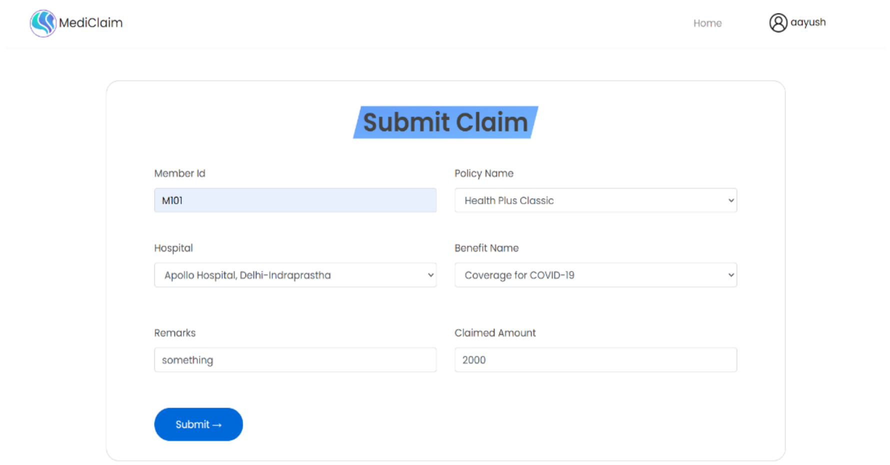




Login Portal

Username: aayush

Password: ••••••••



Submit Claim

Member Id: M101

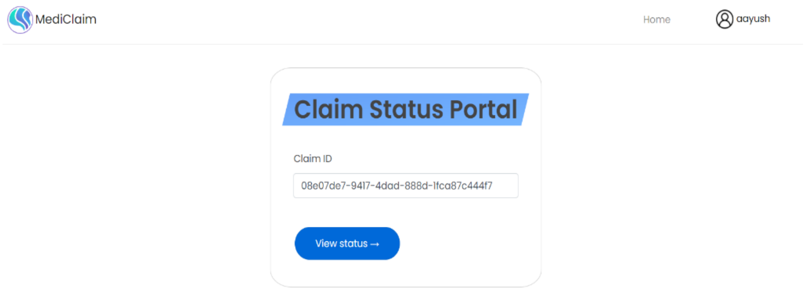
Policy Name: Health Plus Classic

Hospital: Apollo Hospital, Delhi-Indraprastha

Benefit Name: Coverage for COVID-19

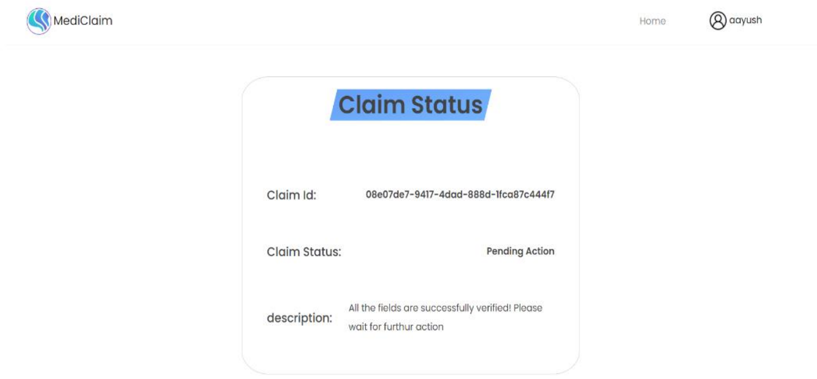
Remarks: something

Claimed Amount: 2000



Claim Status Portal

Claim ID: 08e07de7-9417-4dad-888d-1fca87c444f7



Claim Status

Claim Id: 08e07de7-9417-4dad-888d-1fca87c444f7

Claim Status: Pending Action

description: All the fields are successfully verified! Please wait for furthur action

Chapter 05: CONCLUSION

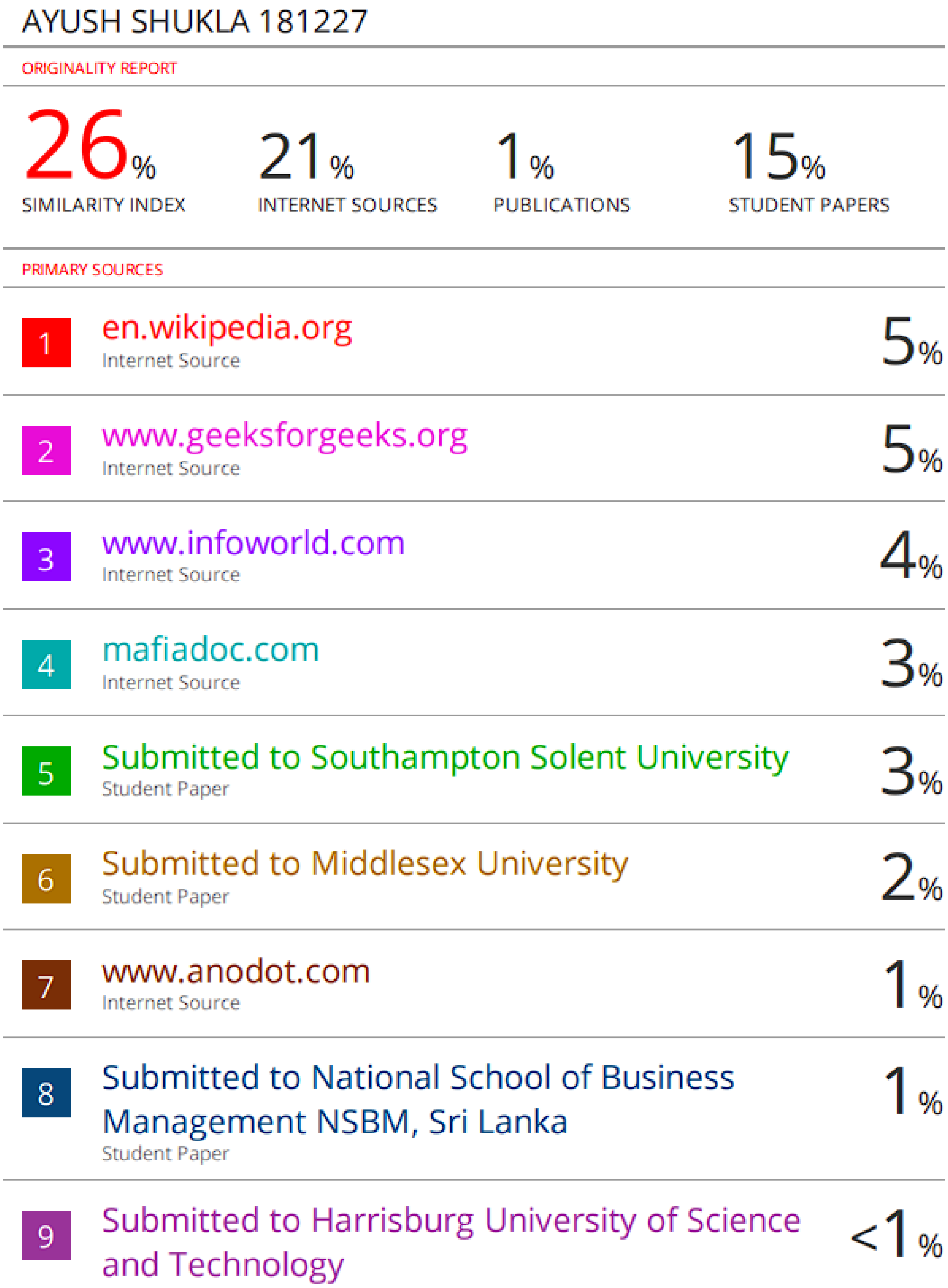
5.1 Conclusion

In conclusion, this internship has become a wonderful and rewarding experience. I can say that my stay with Cognizant India (P) Ltd - A Paymentus Company was quite beneficial to me. Needless to say, the technical components of my work aren't perfect and, given enough time, could be improved. As someone who had no prior experience with full stack development, I think the effort I spent learning and understanding about was well spent, as it helped to construct a fully effective web service. Two of the most significant lessons I've learned are time management and self-motivation.

REFERENCES

Java 8 Parallel Programming	https://dzone.com/articles/parallel-and-asynchronous-programming-in-java-8
Feign client	https://dzone.com/articles/Microservices-communication-feign-as-rest-client
Swagger	https://dzone.com/articles/centralized-documentation-in-Microservice-spring-b
ECL Emma Code Coverage	https://www.eclipse.org/community/eclipse_newsletter/2015/august/article1.php
Lombok Logging	https://javabydeveloper.com/lombok-slf4j-examples/
Spring Security	https://dzone.com/articles/spring-boot-security-json-web-tokenjwt-hello-world
H2 In-memory Database	https://dzone.com/articles/spring-data-jpa-with-an-embedded-database-and-spring-boot https://www.baeldung.com/spring-boot-h2-database
AppInsights logging	https://www.codeproject.com/Tips/1044948/Logging-with-ApplicationInsights
Error response in WebApi	https://stackoverflow.com/questions/10732644/best-practice-to-return-errors-in-asp-net-web-api
Read content from CSV	https://stackoverflow.com/questions/26790477/read-csv-to-list-of-objects
Access app settings key from appSettings.json in a .Net core application	https://www.c-sharpcorner.com/article/reading-values-from-appsettings-json-in-asp-net-core/ https://docs.microsoft.com/en-us/aspnet/core/fundamentals/configuration/?view=aspnetcore-3.1

PLAGIARISM REPORT



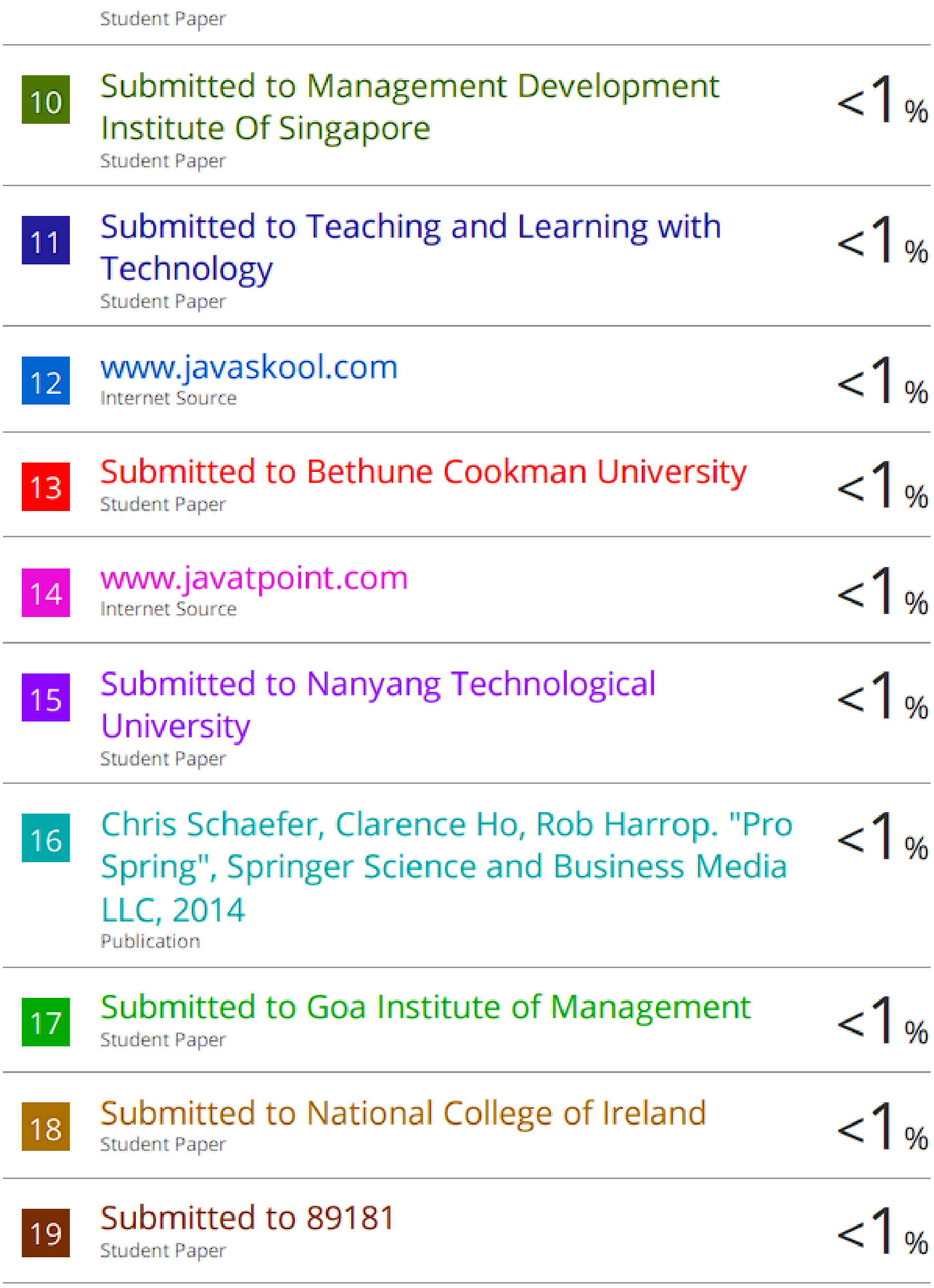
AYUSH SHUKLA 181227

ORIGINALITY REPORT

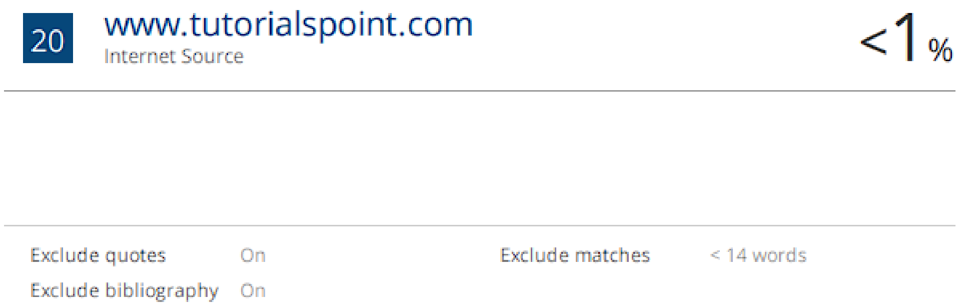
26%	21%	1%	15%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	en.wikipedia.org Internet Source	5%
2	www.geeksforgeeks.org Internet Source	5%
3	www.infoworld.com Internet Source	4%
4	mafiadoc.com Internet Source	3%
5	Submitted to Southampton Solent University Student Paper	3%
6	Submitted to Middlesex University Student Paper	2%
7	www.anodot.com Internet Source	1%
8	Submitted to National School of Business Management NSBM, Sri Lanka Student Paper	1%
9	Submitted to Harrisburg University of Science and Technology	<1%



	Student Paper	
10	Submitted to Management Development Institute Of Singapore Student Paper	<1 %
11	Submitted to Teaching and Learning with Technology Student Paper	<1 %
12	www.javaskool.com Internet Source	<1 %
13	Submitted to Bethune Cookman University Student Paper	<1 %
14	www.javatpoint.com Internet Source	<1 %
15	Submitted to Nanyang Technological University Student Paper	<1 %
16	Chris Schaefer, Clarence Ho, Rob Harrop. "Pro Spring", Springer Science and Business Media LLC, 2014 Publication	<1 %
17	Submitted to Goa Institute of Management Student Paper	<1 %
18	Submitted to National College of Ireland Student Paper	<1 %
19	Submitted to 89181 Student Paper	<1 %



20	www.tutorialspoint.com Internet Source	<1 %

Exclude quotes	On	Exclude matches	< 14 words
Exclude bibliography	On		