

Cognizant Internship Report

Internship report submitted in partial fulfillment of the
requirement for the degree of Bachelor of Technology

in

**Computer Science and Engineering/Information
Technology**

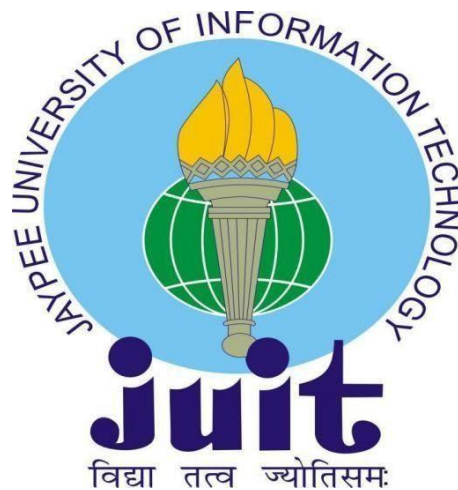
By

Vishrut Sharma (181392)

Under the supervision of

Dr. Sumedha Arora

To



Department of Computer Science & Engineering and
Information Technology

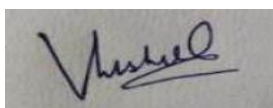
**Jaypee University of Information Technology Wanknaghat,
Solani-173234, Himachal Pradesh**

Candidate's Declaration

I hereby declare that the work presented in this report entitled “ **Cognizant Internship Report**” in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Wanknaghat is an authentic record of my own work carried out over a period from August 2015 to December 2015 under the supervision of **Dr. Sumedha Arora**

(Assistant professor Senior grade Computer Science Engineering).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.



Vishrut Sharma, 181392

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Sumedha Arora
Assistant Professor(S.G.)
Computer Science Engineering
Dated: 25-05-2022

ACKNOWLEDGEMENT

To begin with, I want to express my heartfelt gratitude to almighty God for His divine blessing, which has enabled us to successfully complete the project work.

Supervisor Dr. Sumedha Arora, Assistant Professor (SG), Department of CSE Jaypee University of Information Technology, Waknaghat, deserves my deepest gratitude. My supervisor's deep knowledge as well as his relentless patience, scholarly guidance, persistent and enthusiastic supervision, constructive criticism, insightful advice, and reading many inferior draughts and correcting them at all stages.

I'd like to express my heartfelt gratitude to Dr. Sumedha Arora of the Department of CSE for his invaluable assistance in completing my thesis.

I'd also like to express my gratitude to everyone who has assisted me in making this project a success, whether directly or indirectly. In this specific circumstance, I'd like to express my gratitude to the numerous staff members, both teaching and non-teaching, who have provided me with valuable assistance and facilitated my project.

Finally, I must express my gratitude for my parents' unwavering and unflagging love, support and utmost patience in putting up with us.

Vishrut Sharma (181392)

TABLE OF CONTENTS

Chapter Name	Page No.
Candidate's Declaration	i
Acknowledgement	ii
List of Abbreviations	v
List of Figures	vi
Chapter1: Introduction	
1.1 Cognizant	1
1.2 Digital Technologies	3
1.3 Internship Learnings	5
Chapter 2: OOPS & Data Structures	
2.1 Object Oriented Programming	7
2.1.1 Structure of OOPs	7
2.1.2 Features of OOPs	7
2.2 Data Structures	8
Chapter 3: System Development	
3.1 Introduction to Manual Testing	12
3.1.1 Software Development Life Cycle(SDLC)	12
3.1.2 Testing in Waterfall Model	13
3.1.3 Testing in V-Model	14
3.1.4 Different levels of Software Testing	15
3.1.5 Testing Types	17
3.2 .Introduction: Automation Testing	19
3.2.1 Automation Testing	20
3.3 Selenium	20
3.3.1 Selenium Web Driver	21
3.3.2 Elements in Selenium	22

Chapter 4: SQL Server

4.1 Introduction to RDBMS	27
4.2 Structured Query Language (SQL)	27
4.2.1 Application of SQL	27
4.2.2 SQL Components	28
4.2.3 SQL Architecture	29
4.3 Stored Procedures	30
4.4 Common Table Expressions (CTE)	31
4.4.1 Recursive CTE	31
4.5 Pivot	32
4.6 Dynamic SQL	32
4.7 SQL Server Management Studio	33
4.8 Object Relational Model (ORM)	34
4.9 Entity Framework Core	34
4.9.1 Entity Framework Core Approaches	35
4.10 EF Core Database Providers	36

Chapter 5: Front-End

5.1 HTML	37
5.2 CSS	38
5.3 JavaScript	39
5.3.1 AJAX	39
5.4 TypeScript	40

Chapter 6 Conclusion

References	43
------------	----

LIST OF ABBREVIATIONS

SQL- Structured Query Language

EF - Entity Framework

RDBMS - Relational Database Management System

CTE - Common Table Expression

SDLC- Software Development Life Cycle

LIST OF FIGURES

- Figure 1:** Sample Submission File
- Figure 2:** Paxcom Timeline
- Figure 3:** How Paymentus Works
- Figure 4:** Stack Representation
- Figure 5:** Queue Representation
- Figure 6:** Enque
- Figure 7:** Deque
- Figure 8:** Linked List
- Figure 9:** Time Complexity of Linked List
- Figure 10:** C# Logo
- Figure 11:** LINQ Architecture
- Figure 12:** Generics Code
- Figure 13:** Delegates
- Figure 14:** Program.cs file
- Figure 15:** Customer Class
- Figure 16:** Data Model
- Figure 17:** ProcessInitiator.cs
- Figure 18:** Tools.cs
- Figure 19:** UserInterface.cs
- Figure 20:** Output
- Figure 21:** .NET Framework
- Figure 22:** Architecture of .NET Framework
- Figure 23:** Web API
- Figure 24:** API Architecture
- Figure 25:** N Layer Architecture
- Figure 26:** Customer Controller
- Figure 27:** Customer BLL
- Figure 28:** Customer DAL
- Figure 29:** Startup.cs
- Figure 30:** CustomerBookings Table
- Figure 31:** Customers Table
- Figure 32:** Http Client Server Model

Figure 33: Http Methods

Figure 34: Postman

Figure 35: SQL Components Processing

Figure 36: Stored Procedure

Figure 37: CTE

Figure 38: Recursive CTE

Figure 39: Pivot

Figure 40: SQL Server Management Studio Environment

Figure 41: Model of Code First Approach

Figure 42: Model Of Database First Approach

Figure 43: Design of Customer Table

Figure 44: Customer Class automatically generated by EF Core

Figure 45: EF Core Dependencies

Figure 46: HTML Page Structure

Figure 47: FlexBox Structure

Figure 48: Compilation process of TypeScript

CHAPTER 1: INTRODUCTION

1.1 Cognizant



Figure 1: Cognizant Logo

Cognizant (Nasdaq-100: CTSB) is one of the world's driving proficient administrations organizations, changing clients' business, working and innovation models for the advanced time. Our novel industry-based, consultative methodology helps a considerable lot of the most popular associations in each industry also, topography imagine, construct and run more inventive and effective organizations.

Established in 1994 as an innovation improvement arm of The Dun and Bradstreet Corporation, we were veered off as a free organization in 1996, and have worked intimately with enormous associations to assist them with building more grounded organizations from that point forward. Today, Cognizant delivers engineering solutions to modern organizations to work on regular day to day existence, assisting a portion of the world's most settled organizations with staying the most cherished brands.

In the present quick changing innovation scene, we work with our clients to propel each part of how they serve their clients: digitizing their items, administrations and client encounters; computerizing their business processes; and modernizing their innovation foundations.

Set forth plainly, we assist clients with bridling advances to address their everyday necessities and keep their organizations important. As the accomplice they go to execute on their computerized needs, we center around IoT, AI, programming and cloud — the innovations that are changing the idea of business.

Recognized as a leader

Ranked #185 on the **Fortune 500** (June 2021)

Ranked #523 in **Forbes Global 2000** (May 2021)

Ranked #327 in **Forbes 2021 World's Best Employer list** (Oct 2021)

Ranked #350 in **Forbes World's Best Employers for Diversity**
(April 2021)

Ranked #113 in **Fortune's World's Most Admired Companies**
(Feb 2022)

Figure 2: Cognizant as industry leader

Today, making esteem by utilizing innovation is very industry-explicit, so we keep on developing our ability in 20 distinct enterprises, including banking and monetary administrations, medical care, assembling and retail. Also, to assist with speeding clients' excursions toward becoming computerized, we bring our advanced capacities and industry aptitude together into level contributions and industry arrangements that speed up the most fundamental jumps that the present innovation makes conceivable, and supplement those arrangements with counseling and administrations worked for the speed of business today.

We think of it as our obligation to cause individuals to feel comfortable later on, regardless of how innovation becomes empowered. Along these lines, we are focused on assisting with addressing a portion of humanity's generally troublesome challenges in a manner that is useful and agreeable for individuals through the work we do, and through putting resources into preparing individuals all over the planet in the computerized abilities that will be expected to accomplish that work. With base camp in the US and a quickly growing impression that reaches out from India and China to Europe, North America, South America and the Middle East, we're focused on building computerized ability from one

side of the planet to the other so everybody can profit from the full range of human creativity. We work together locally with clients, face to face and in their nearby dialects.

We accept that the open door introduced by innovation has never been more noteworthy, and in light of that open door, Cognizant will keep on teaming up with clients to modernize their organizations, making day to day existence far and away superior for them, their clients and the networks they serve.

1.2 Digital technologies

Innovation is vital to Cognizant on the grounds that it means a lot to our clients. What's more, by innovation, we don't mean just UIs or advanced promoting. As far as we might be concerned, innovation implies industry-explicit arrangements that are woven profoundly into our clients' organizations to make new degrees of significant worth, enterprise wide. That is the reason we're centered around utilizing information, cloud, IoT and computer programming — and teaming up intimately with our industry organizations in joining forces with all aspects of our clients' associations from front to back, at scale.

Keeping that in mind, we consistently put resources into propelling our aptitude in the strategic advancements our clients need to flourish:

- **IoT:** We plan and assemble clever arrangements that assist our clients with opening worth in their items, tasks and administrations. That is all that from ongoing virus bind observing to guard food and medication, to forestalling water and fire debacles with canny siphons.
 - IoT and Engineering market pioneer in assembling, life sciences and retail
 - Demonstrated gas pedals drive IoT development 5x quicker
 - Idea to guide in 30 days.
- **AI:** We are business engineers who assist clients with modernizing their information so they can pursue the most ideal choices conceivable, with speed and certainty at the least gamble. From joining forces with XPRIZE to securely return society and breaking point COVID's monetary effect, to smoothing out disease drug

improvement — our specialists bridle the force of information and insight to tackle a portion of the world's most difficult issues.

- Further develop direction: 10x quicker
- Increment readiness: strong choices utilizing 1/twentieth of the information
- Speed time to showcase: 30%-60% quicker than contenders

- **Software Engineering:** We're reason driven. We engineer programming that addresses human issues across a scope of ventures. Our designers have a genuine effect — whether we're driving a proficient harvest water system or making a telemedicine application that makes medical services open.

- One of the best three computerized designing administrations suppliers around the world
- Slice improvement time down the middle while bringing down risk with our devices

- **Cloud:** We enable and extend our clients' development potential with cloud by contracting information cycles from three days to three minutes.

- 100+ in-house plans and gas pedals
- 600+ cloud-drove change projects
- 25,000+ business-basic applications relocated

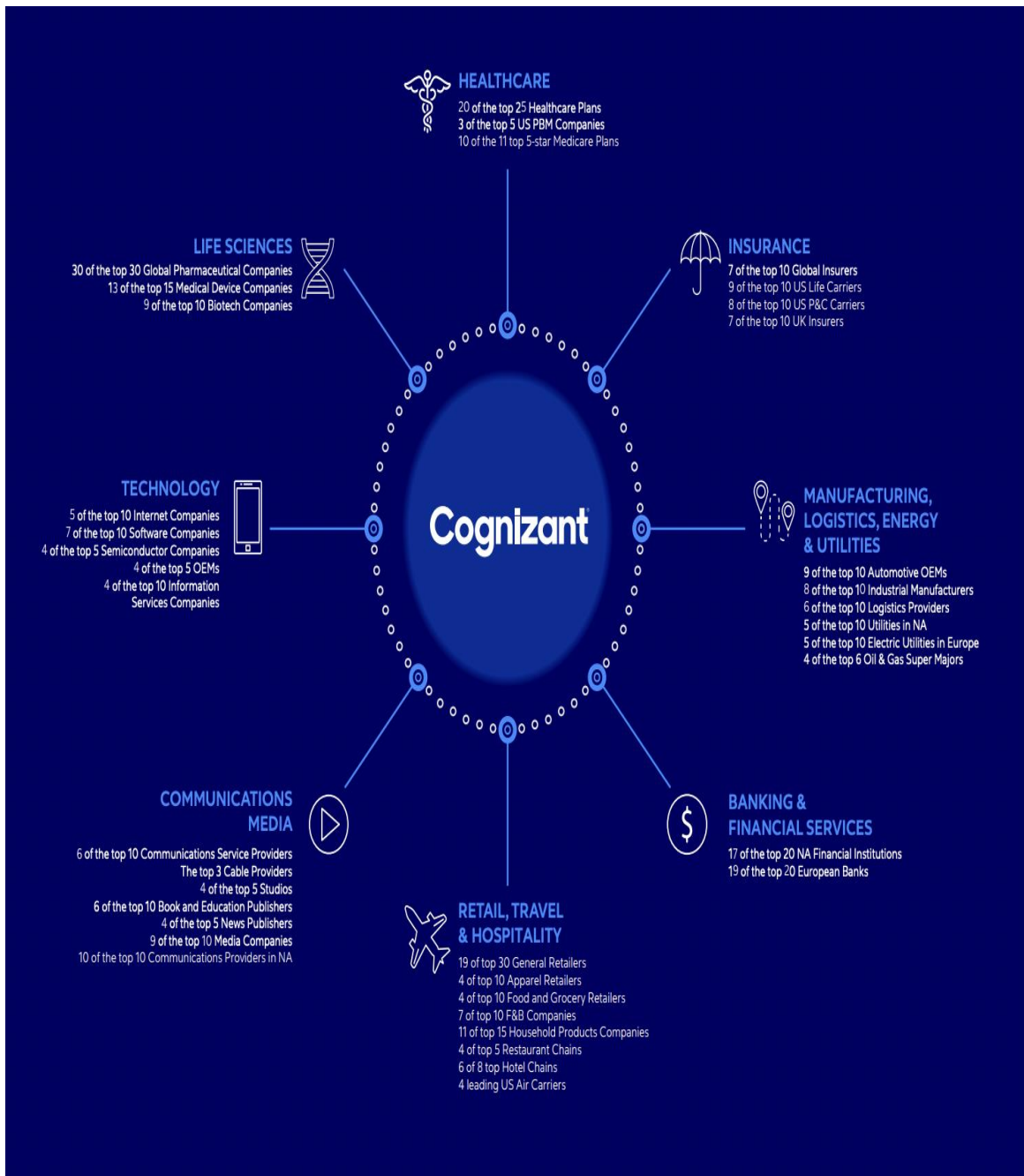


Figure 3: Cognizant and its clients

1.3 Internship learnings:

Cognizant is a New Jersey based software company providing information technology services and consulting. I worked at Cognizant Technology Solutions India Pvt Ltd at Hinjewadi, Pune branch. My internship started and was put into the Quality

assurance(QEA) domain under the Software Development Engineer in Test (SDET) role. I also learned soft skills like communicating within a corporate firm and working with a team. I was successfully able to understand various basics and advanced concepts related to both development and testing.

The working culture of the company is great. I thoroughly enjoyed myself working there. Cognizant has a typical blend of work and fun. Although I didn't get to spend any time in the company office and started working from home after the coronavirus pandemic lockdown, I really enjoyed the online sessions and events conducted at regular events by the company. And even after the lockdown the communication between me and my team was good through various meeting platforms like skype and google meets.

There are numerous things I love about my internship - the experience/information I've acquired, the balance between fun and serious activities. In any case, in the event that I needed to put it on a certain something, what I like most is, at last, the way that this chance truly opened the entryway into my profession. The most stunning thing about this temporary position experience is that it truly overcame any barrier between learning things and really applying a portion of this information into a reality, and getting paid for it! The way that I presently have an entry-level position straightforwardly identified with what I concentrate on makes me study more diligently and makes me need to sort out how I can apply a greater amount of what I'm realizing in an internship. It's interesting, in light of the fact that things I've learned in college help me at work.

CHAPTER 2 : OOPs AND DATA STRUCTURES

2.1 Object Oriented Programming

Object-oriented programming (OOP) is a programming paradigm which does not have a major focus on functions and logic, but is more concerned about the functionality that revolves around data and objects.

It merely focuses on the logic required for the management, but strongly on the objects. For the projects and ideology that is large and is prone to update, the developers prefer this method of programming. If the work is concerned with the production systems, designing, or mobile applications, the developers aim at working on the principles and features of OOPs.

2.1.1 Structure of OOPs

1. **Classes:** It is just a blueprint of the methods, attributes and objects. It is considered to be a logical entity and used to define the states of variables, methods and attributes.
2. **Objects:** Object is an actual instance of a class that implements the traits of the methods and attributes of the class. It is an entity that does not exist logically but has a structure and an existence.
3. **Methods:** These are the functions inside a class to describe the behaviors of objects. They are meant to perform certain tasks.
4. **Attributes:** Attributes are needed to store the information and data. Inside a class, they are used to represent the state and quality of the objects.

2.1.2 Features of OOPs

1. **Encapsulation:** As the name suggests, encapsulation means to put the content of the class into a capsule. The important information is bound inside it and all the unnecessary details are hidden from the user.

a) Data Binding: Binding the information, attributes, methods and data inside a class.

b) Data Hiding: Hiding the unnecessary details from the user.

Its major focus is to provide security.

2. Abstraction: This is a feature which is intended to not disclose the backend functionality and details to the user, as it is totally not needed.

3. Inheritance: Inheritance is a feature by virtue of which classes can inherit the properties of some other classes. The classes which inherit the properties are called the derived classes and whose properties are inherited are called the base classes.

4. Polymorphism: Polymorphism is the process by which a piece of code, data, technique, or object acts differently in different situations.

a) Compile-time Polymorphism: Compile time polymorphism, commonly referred to as static polymorphism, is a sort of polymorphism that occurs during the compilation process. The compiler determines the shape or value that the entity in the image must take.

b) Run-time Polymorphism: Runtime polymorphism, also known as Dynamic Polymorphism, is a sort of polymorphism that occurs while the programme is running. It signifies that the compiler cannot decide. As a result, the execution determines what shape or value must be taken.

2.2 Data Structures

Data Structures is a medium of storing data and organizing it. It is a way of arranging data on the system in order to have an efficient access and update. During the internship, before we could actually move to the major work ahead, our mentor wanted us to brush up our skills on Data Structures.

During the course of our proceedings we were not asked to go through all the Data Structures, just a few of them that were needed for our further meetings.

1. Arrays: Arrays are the Data Structures that enable contiguous memory allocation. It follows an index based allocation, which makes it easier to access an element. All the elements have the same data type.

2. Stack: Stack are the special Data Structures that follow the Last In First Out principle.

It is a linear data structure.

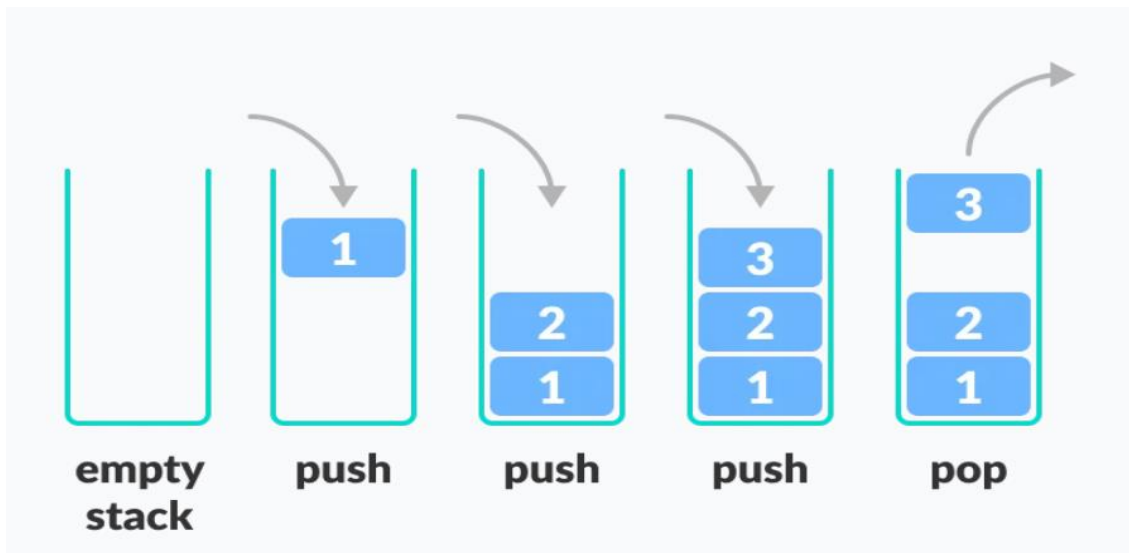


Figure 4: Stack Representation

3. Queue: Queue Data Structure follows the First In First Out principle. It is a linear Data Structure.

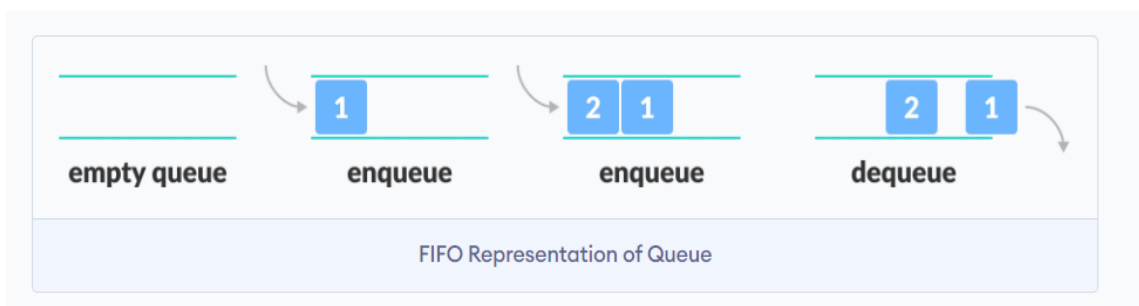


Figure 5: Queue Representation

The two operations are :

a) Enqueue

b) Dequeue

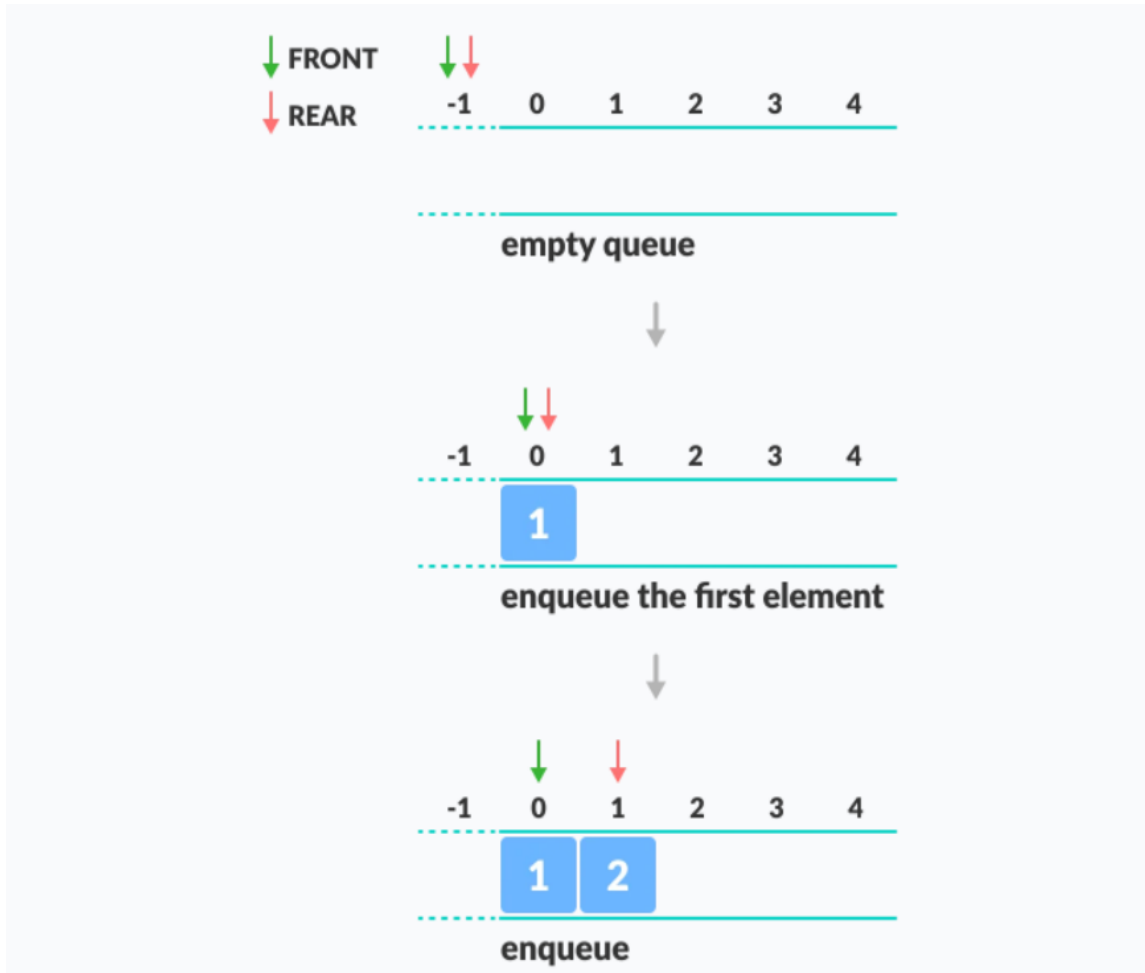


Figure 6: Enque

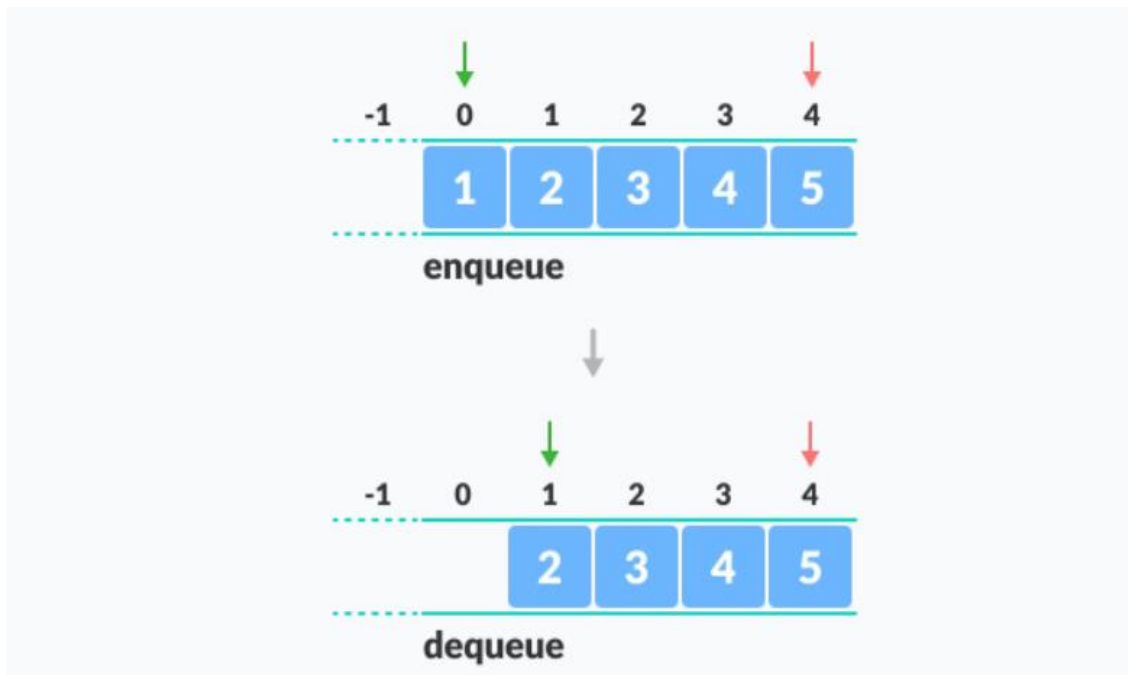


Figure 7: Deque

4. **Linked List:** This is a linear Data Structure in which elements are connected to each other in a series of nodes. A node contains two important pieces of information, the value and the address of the next node.

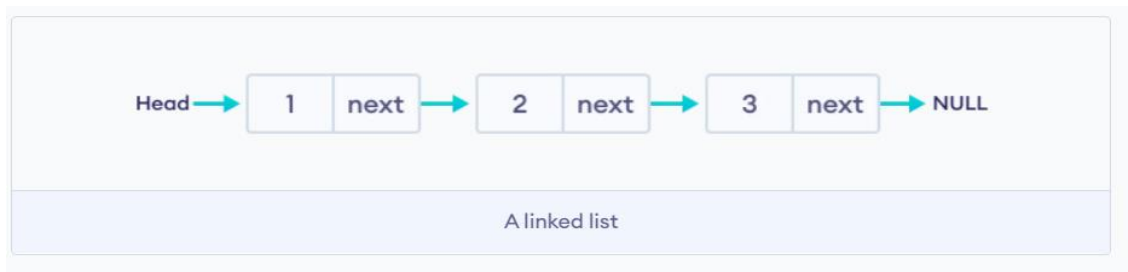


Figure 8: Linked List

Time Complexity

	Worst case	Average Case
Search	$O(n)$	$O(n)$
Insert	$O(1)$	$O(1)$
Deletion	$O(1)$	$O(1)$

Figure 9: Time Complexity of Linked List

Chapter 3: Manual and Automation Testing

3.1 Introduction to Manual Testing :

Manual testing is the cycle where the imperfections are distinguished, confined, oppressed for avocation and guarantee that the item is without imperfection, to create quality items. Anyway it requires important information like about various sorts of manual testing, software development life cycle. In simple words, manual testing can be defined as a type of testing which is done without the use of automation tools such as Selenium. Although it is a primitive technique of finding bugs, it proves to be very useful and helpful in certain test scenarios. One of the Software Testing Fundamental is "100 percent Automation is beyond the realm of possibilities' '. This makes Manual Testing imperative. Any new application should be manually tested before it can be automated with tools such as Java and Selenium.

3.1.1 Software Development Life Cycle(SDLC):

A Software Development Lifecycle Model portrays the kinds of exercises performed at each stage in a product improvement venture, and how the exercises connect with each other sensibly and sequentially. SDLC can be seen to be classified into two types mainly which are sequential software development models and secondly, iterative and incremental development models.

- **Sequential Software Development:** A successive improvement model portrays the product advancement process as a direct, consecutive progression of exercises. This implies that any stage in the improvement cycle ought to start when the past stage is finished. In principle, there is no crossover of stages, however practically speaking, having early criticism from the accompanying phase is useful.
- **Iterative and Incremental Software Development:** The fundamental thought behind this strategy is to foster a framework through rehashed cycles (iterative) and in more modest bits all at once (gradual). Iterative and Incremental improvement is a blend of both iterative plan or iterative technique and steady form model for improvement. "During programming improvement, more than one emphasis of the product advancement cycle might be in the works simultaneously." In this gradual model, the entire necessity is partitioned into different forms. During every

emphasis, the improvement module goes through the prerequisites, plan, execution and testing stages.



Figure 10: Software Development Lifecycle

3.1.2 Testing in Waterfall Model:

The Waterfall model is the earliest SDLC approach that was utilized for programming advancement. This type of Model represents the product improvement process in a linear sequential flow. In this waterfall model, the stages don't overlap. This implies that any stage in the improvement cycle starts provided that the past stage is finished.

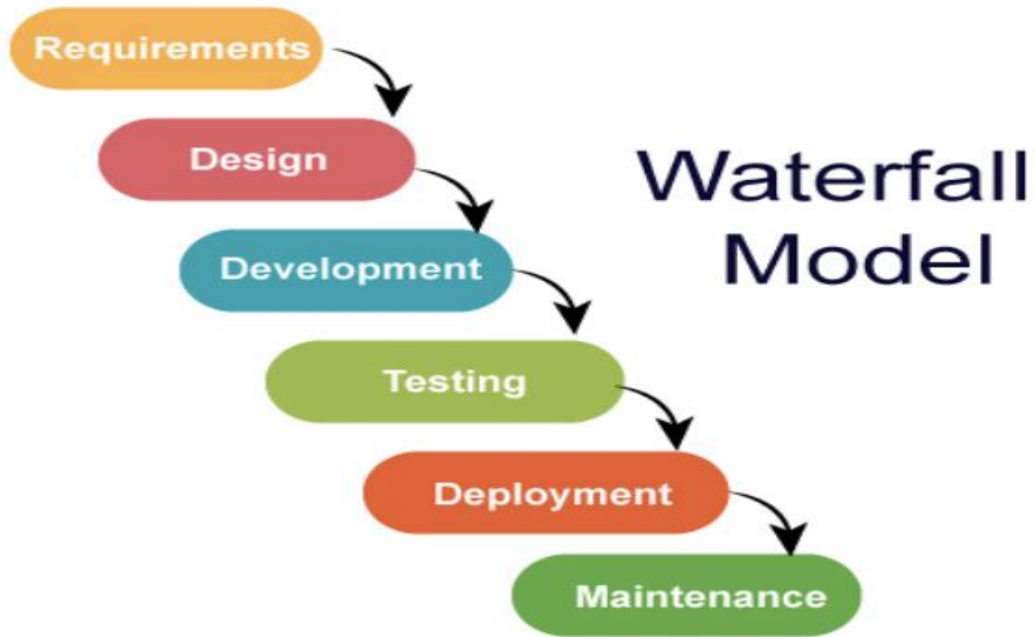


Figure 11: Testing in Waterfall Model

3.1.3 Testing in V-model :

The V-model is a SDLC model where execution of cycles occurs in a successive way in a V-shape. It is otherwise called the Verification and Validation model. The V-Model is an augmentation of the waterfall model and depends on the relationship of a testing stage for each related improvement stage. This intends that for each and every stage in the improvement cycle, there is a straightforwardly related testing stage. This is an exceptionally restrained model and the following stage begins solely after fulfillment of the past stage. Dissimilar to the Waterfall model, the V-model incorporates the test cycle all through the advancement interaction, executing the standard of early testing.

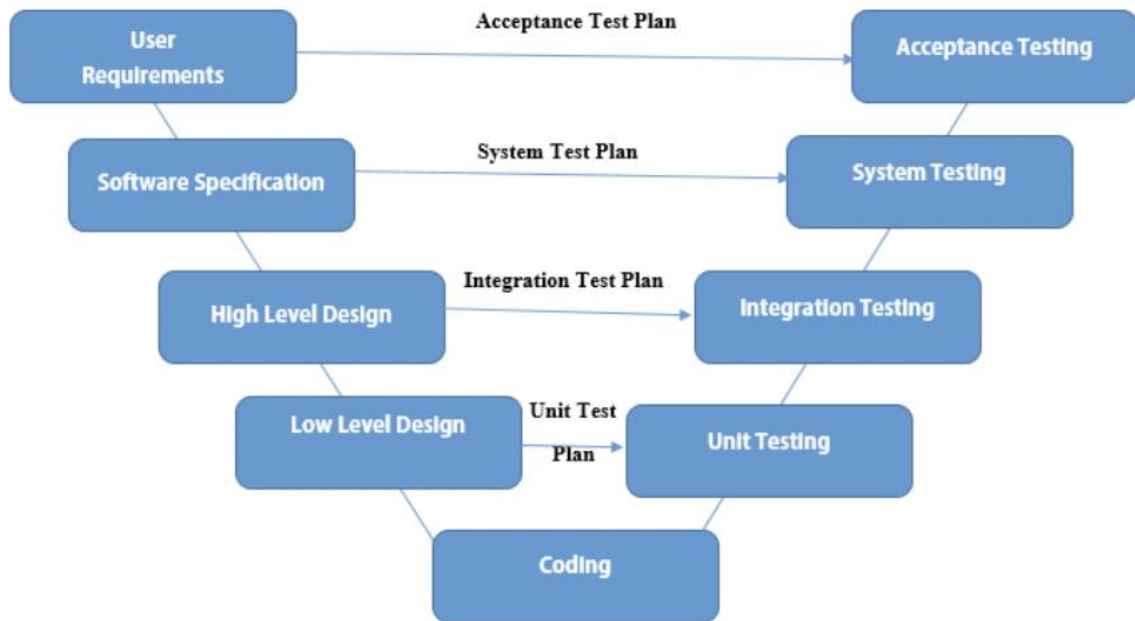


Figure 12: Testing in V-model

3.1.4 Different Levels of Software Testing:

- Unit Testing:** A unit is the littlest conceivable testable programming part. A unit is little, so it is more straightforward to configure, execute, record, and dissect test results than bigger pieces of code are. Bugs uncovered by a unit test are not difficult to find and somewhat simple to fix. In computer programming, unit testing is a product testing technique by which individual units of source code are tried to decide if they are good for use. aims to verify each part of the software by isolating it and then perform tests to demonstrate that each individual component is correct in terms of fulfilling requirements and the desired functionality. This type of testing is done by developers.
- Integration Testing:** This type of testing is done by developers. In this testing the main goal is to test various pieces of the framework in a mix to survey assuming they work accurately together. By testing the units in gatherings, any issues in the manner in which they communicate together can be recognized. Incorporation testing is a kind of testing intended to check the blends of various units, their connections, the manner in which subsystems join into one normal framework, and code consistency with the prerequisites.
- System Testing:** all the components of the software are tested as a whole in order to ensure that the overall product meets the requirements specified. It is done by the testers. Framework testing is a kind of testing where the application overall is tried for its consistency with useful and non-practical prerequisites. It is done by the QA group after the consummation of mix testing and before the last acknowledgment

testing. Since in framework testing the entire framework is tried from start to finish and the information on coding or inside design isn't needed. Thus, framework testing goes under Black-box Testing. The framework is tried all in all in this period of testing. Indeed, even after effective unit and coordination testing, numerous complicated situations might have issues unseen by prior testing stages. Framework testing helps in uncovering those imperfections. The framework is assessed based on the client's perspective, which is a significant method for grasping the end-clients prerequisite. All things considered, they will utilize the framework at last.

- **Acceptance Testing:** Acceptance testing is the last degree of programming testing done by the users where the framework is tried for consistency to its business prerequisites. It is performed by the client or the end clients with the goal to check whether the item is good for conveyance. Before acknowledgment testing, the application has been tried by the QA group for example Inner testing group. The QA group will test, and designers will foster the application in view of the necessity archives given to them. They might have their own comprehension and might interpret the necessities because of an absence of space information. It is conceivable that their comprehension is not quite the same as that of business clients. During acknowledgment testing, business clients get an opportunity to check in the event that everything matches their assumptions. During acknowledgement testing, business clients get to see the end result. Clients can check whether the framework works as indicated by the given necessities. UAT likewise guarantees that the prerequisites have been conveyed and executed actually. Business clients can acquire trust in showing the application in the market for example to the end-clients. In this manner, criticism/ideas given during acknowledgment testing can be useful in the following deliveries.



Figure 13: Levels of Software Testing

3.1.5 Testing Types :

- **Functional Testing:** is a type of testing which verifies that each function of the software application operates in conformance with the requirement specification. It tests what the system does. Functional testing is a kind of testing which includes testing the useful necessities of the framework under test. Every one of the application's business prerequisites details is approved in this sort of testing by breezing through assessment information and contrasting the genuine outcome and the normal outcome. Functional testing as the name proposes manages testing of the various functionalities of the application under test.
- **Non-Functional Testing:** is a type of testing to check non-functional aspects (performance, usability, reliability, etc.) of a software application. It tests how well the system performs. Example:
 - In functional testing, we test the login functionality, does it work like expected or not?
 - In non-functional testing we may test the performance of the system when 100 users login simultaneously.
- **Black-Box Testing:** is a product testing technique in which the inner construction/plan/execution of the thing being tried isn't known to the analyzer . Black box testing is the sort of testing where an application is tried in light of its necessities determinations without the requirement for information on its inside engineering. It tests both practical as well as non-useful necessities of the application. Information/admittance to the coding/plan/inner engineering of the product isn't needed. Following are the kinds of black-box testing methods:
 - **Equivalence class partitioning-** includes apportioning the info information into coherent gatherings or equality classes. Every one of the information things lying in a proportionality class are thought to be handled similarly by the application when passed as information. For example, for programming that finds the square of a number, we can have different comparability classes like - every single positive number, negative numbers, decimal numbers, negative decimal numbers, and so on. Its benefit is - the general test execution time lessens as the quantity of test information enormously diminishes.
 - **Boundary value analysis-** is a black-box testing procedure, firmly connected with identicalness class parceling. In this procedure, we examine the way of behaving of the application with test information dwelling at the limits of the equality classes. Its benefit is - it is simpler and quicker to find surrenders as the thickness of imperfections at limits is more. For example for proportionality classes with a contribution between 0 to 100, the test information utilizing boundary value analysis would be 0 and 100.

Black Box Testing

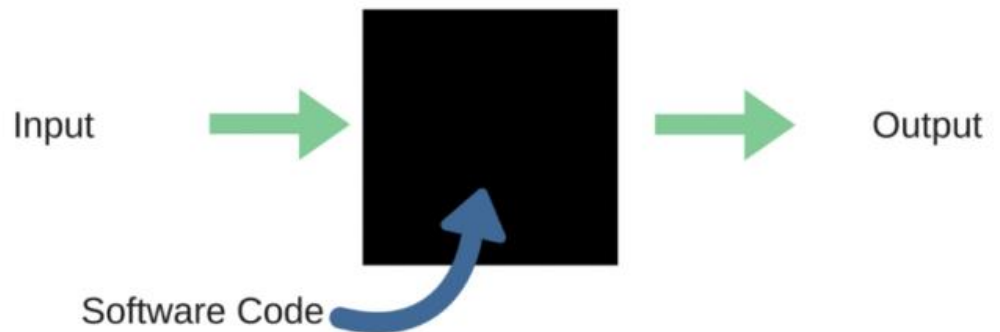


Figure 13: Black Box Testing

- **White-Box Testing:** White box testing is likewise referred to as structure based testing and glass box testing. In this procedure, the information on code or inside engineering of the framework is expected to do the testing. The application's plan, design, and source code are expected to be read up for this kind of testing. Since engineers as of now know about every one of these. Subsequently, for the most part the designers are expected to do white-box testing methods to test their own code. It is material to bring down degrees of testing types - unit testing and coordination testing. Where we need to manage individual modules of the application and their communication. Following are the kinds of white-box testing methods:
 - **Statement testing**-In statement testing, the test scripts are intended to execute the code. The inclusion is estimated by the line of code or statements executed by test scripts.
 - **Decision testing**- also known as branch testing. In decision testing, we measure the level of decision focus. For instance, the aggregate if-else conditions executed out of the absolute decision focuses on the application.

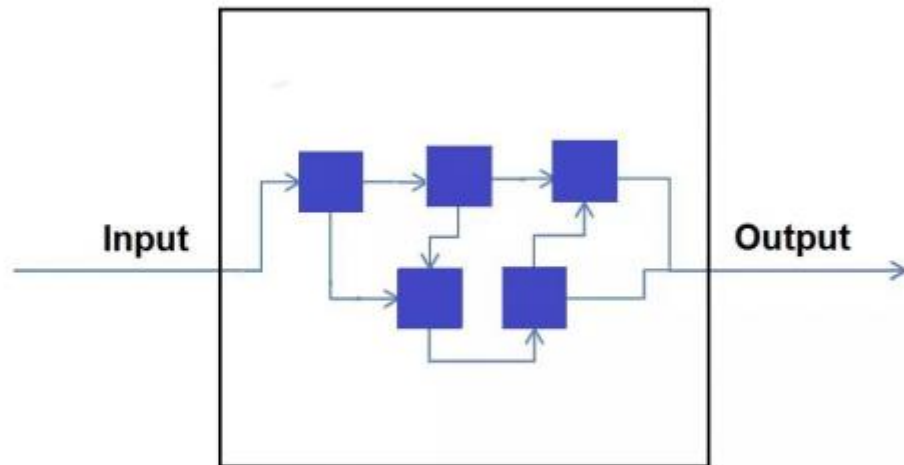


Figure 14: White Box Testing

3.2 Introduction : Automation Testing

Although automated testing and manual testing both have contributed fundamentally in guaranteeing an item's quality effectively, with additional organizations picking the spry style of improvement, the degree for test computerization has most certainly gone up. There is a rising interest for fresher regions to be automated, more up to date devices to work on the interaction and more test architects to be prepared for the gig. A test robotization cycle ought to continuously be an arranged one, any other way, it will wind up in being a mammoth suite which is challenging to keep up with and make due.

Just a broad testing cycle can guarantee the nature of a framework and to this end programming testing administrations are extremely popular. Testing starts with developments and making new aspects and rethinking. New testing cycles and techniques are made to satisfy the needs and surpass the assumptions. We should recognize the expected areas of computerization and aspects of effective Automation Testing.

3.2.1 Automation Testing:

The results of each and every product improvement bunch are tried however they might have abandons. Test engineers endeavor to get the product defects before the item is delivered. However, even with the best of manual testing processes they frequently appear again. An automated testing can playback pre-recorded and predefined activities, make correlation of the outcomes to expected conduct and report disappointment or progress of manual tests to a test engineer. The computerization tests, once made, can be rehashed effectively and they can be widened to do those errands which are unimaginable with manual testing. To this end automated programming testing is an essential component of any fruitful improvement project. It is the most ideal way to expand the proficiency, inclusion and viability of your product testing.

3.3 Selenium:

Selenium is one of the most prestigious open-source test robotization structures. Selenium permits test robotization of web-applications or sites across various programs and working frameworks. Selenium offers similarity with various programming dialects like Java, JavaScript, Python, C# from there, the sky's the limit, permitting analyzers to robotize their site testing in any programming language that they are OK with.

Utilizing the Selenium system, analyzers can convey test cycles quicker via robotizing rehashed experiments. When coordinated with CI/CD pipeline, Selenium can likewise assist with a durable, sans bug discharge sending pipeline.

Selenium is a tool:

- It is an open-source device for making test suite
- Assists in testing with webbing application across programs and stages
- It is not a single tool but suite of softwares
- It is a convenient structure for testing web application

Advantages of Selenium:

- Selenium is an open source and versatile Web testing Framework
- upholds different working frameworks, programs and programming dialects

- upholds equal test execution that decreases time and expands the proficiency
- Selenium orders are classified concerning various classes

Selenium Limitations:

- Selenium doesn't uphold work area applications.
- Required help from local area gatherings to get the specialized issues settled
- ought to be aware somewhere around one of the upheld programming dialects
- has no inbuilt detailing ability
- impractical to perform testing on pictures

3.3.1 Selenium WebDriver:

- The test contents can be created utilizing any of the upheld programming dialects and can be run straightforwardly in most present day internet browsers.
- Dialects upheld by WebDriver incorporate C#, Java, Perl, PHP, Python and Ruby.
- A large portion of the orders utilized in Selenium WebDriver are basic and simple to carry out Handles Navigation, Alerts/Dropdowns, Switching between windows, and so on.
- Tracking down Element: It is significant in a website page; Each component has 3 significant properties: id, name and class Should get to HTML labels.
- Selenium WebDriver, involves a bunch of orders for performing various tasks. Orders are just strategies written in Java language

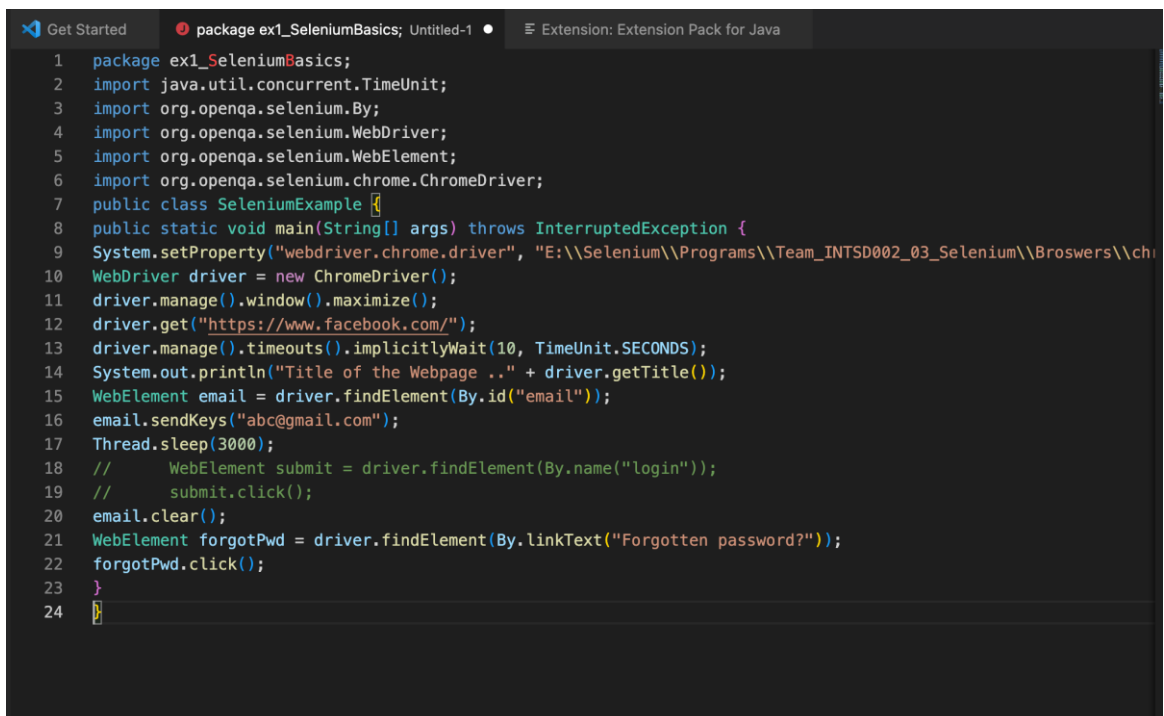
What should the script contain?

- Guarantee significant bundles/classes are imported
- Summon a program
- Make case of the Web Driver
- Amplify the screen window
- Give the URL of the site page and send off the site
- Stand by till the website page loads
- Find each Web Element you need to get to
- Conjure an activity on every one of those Web Element
- At long last close the Web Driver

3.3.2 Elements in Selenium:

Locators:

- **Attribute**
 - **id**
 - **name**
 - **class**
- **Text**
 - **link text**
 - **partial link text**
- **TagName**
 - **tagName**
- **Dynamic Locator-Xpath**
 - **absolute Xpath**
 - **relative Xpath**

A screenshot of an IDE window titled 'package ex1_SeleniumBasics; Untitled-1'. The code is in Java and demonstrates Selenium WebDriver operations. It includes imports for TimeUnit, By, WebDriver, WebElement, and ChromeDriver. The main method sets the ChromeDriver path, maximizes the window, navigates to 'https://www.facebook.com/', waits for 10 seconds, prints the page title, finds an element by ID 'email', sends keys 'abc@gmail.com', sleeps for 3000ms, and then finds elements by name 'login' and link text 'Forgotten password?'.

```
1 package ex1_SeleniumBasics;
2 import java.util.concurrent.TimeUnit;
3 import org.openqa.selenium.By;
4 import org.openqa.selenium.WebDriver;
5 import org.openqa.selenium.WebElement;
6 import org.openqa.selenium.chrome.ChromeDriver;
7 public class SeleniumExample {
8     public static void main(String[] args) throws InterruptedException {
9         System.setProperty("webdriver.chrome.driver", "E:\\Selenium\\Programs\\Team_INTSD002_03_Selenium\\Browsers\\chi
10 WebDriver driver = new ChromeDriver();
11 driver.manage().window().maximize();
12 driver.get("https://www.facebook.com/");
13 driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
14 System.out.println("Title of the Webpage .." + driver.getTitle());
15 WebElement email = driver.findElement(By.id("email"));
16 email.sendKeys("abc@gmail.com");
17 Thread.sleep(3000);
18 //     WebElement submit = driver.findElement(By.name("login"));
19 //     submit.click();
20 email.clear();
21 WebElement forgotPwd = driver.findElement(By.linkText("Forgotten password?"));
22 forgotPwd.click();
23 }
24 }
```

Figure 15: Selenium: basics-locators

```

Get Started package ex1_SeleniumBasics; Untitled-1 Extension: Extension Pack for Java
1 package ex1_SeleniumBasics;
2 import java.util.concurrent.TimeUnit;
3 import org.openqa.selenium.WebDriver;
4 import org.openqa.selenium.chrome.ChromeDriver;
5
6
7 public class NavigateURLs {
8
9 public static void main(String[] args) throws InterruptedException {
10
11 System.setProperty("webdriver.chrome.driver", "E:\\Selenium\\Programs\\Team_INTSD002_03_Selenium\\Browsers\\chromedriver.exe");
12 WebDriver driver = new ChromeDriver();
13 driver.manage().window().maximize();
14 driver.get("https://www.facebook.com/");
15 driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
16 System.out.println("Title of the Webpage .. " + driver.getTitle());
17 System.out.println("Current URL .. " + driver.getCurrentUrl());
18 Thread.sleep(3000);
19 driver.navigate().to("https://www.google.com");
20 System.out.println("Title of the new Webpage .. " + driver.getTitle());
21 System.out.println("New URL .. " + driver.getCurrentUrl());
22 Thread.sleep(3000);
23 driver.navigate().back();
24 System.out.println("Title of the Webpage after navigating back.. " + driver.getTitle());
25 System.out.println("URL of the Webpage after navigating back.. " + driver.getCurrentUrl());
26
27 driver.navigate().forward();
28 driver.navigate().refresh();
29 }
30
31 }

```

Figure 16: Selenium: navigating b/w different URLs

```

Get Started package ex1_SeleniumBasics; Untitled-1 Extension: Extension Pack for Java
1 package ex1_SeleniumBasics;
2
3 import java.util.concurrent.TimeUnit;
4 import org.openqa.selenium.By;
5 import org.openqa.selenium.WebDriver;
6 import org.openqa.selenium.WebElement;
7 import org.openqa.selenium.chrome.ChromeDriver;
8 import org.openqa.selenium.support.ui.Select;
9
10 public class HandleList {
11
12 public static void main(String[] args) throws InterruptedException {
13 System.setProperty("webdriver.chrome.driver", "E:\\Selenium\\Programs\\Team_INTSD002_03_Selenium\\Browsers\\chromedriver.exe");
14 WebDriver driver = new ChromeDriver();
15 driver.manage().window().maximize();
16 driver.get("https://www.roboform.com/filling-test-all-fields");
17 driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
18 WebElement month = driver.findElement(By.name("66mm"));
19 Select selMonth = new Select(month);
20 selMonth.selectByIndex(3);
21 Thread.sleep(3000);
22 selMonth.selectByIndex(7);
23 Thread.sleep(3000);
24 selMonth.selectByValue("11");
25 Thread.sleep(3000);
26 selMonth.selectByVisibleText("Jan");
27 System.out.println("Selected value from dropdown.. " + selMonth.getFirstSelectedOption().getText());
28 // month.click();
29 // month.sendKeys("Jun");
30 // month.click();
31 }
32 }
33
34

```

Figure 17: Selenium: how to handle List element

Xpath:

- A query language used for navigation thru XML document to locate different elements

- It is a strategy to locate elements in Selenium
- Starts with double slash //
- Uses single slash between each condition
- Uses double slash to by-pass parent tags
- For attribute use @
- Its case & space sensitive while searching for a value
- Uses and, or operators to give multiple conditions
- For multiple conditions, use same or different attributes
- Index starts from 1

```

1 package ex2_SeleniumExamples;
2 import java.util.concurrent.TimeUnit;
3 import org.openqa.selenium.By;
4 import org.openqa.selenium.WebDriver;
5 import org.openqa.selenium.WebElement;
6 import org.openqa.selenium.chrome.ChromeDriver;
7
8 public class UseXPath {
9     public static void main(String[] args) {
10        System.setProperty("webdriver.chrome.driver", "E:\\Selenium\\Programs\\Team_INTSD002_03_Selenium\\Browsers\\chromedriver.exe");
11        WebDriver driver = new ChromeDriver();
12        driver.manage().window().maximize();
13        driver.get("https://www.roboform.com/filling-test-all-fields");
14        driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
15        WebElement lName = driver.findElement(By.xpath("//input[@name='04lastname']"));
16        lName.sendKeys("John");
17        //WebElement fullName= driver.findElement(By.xpath("//input[@name='04lastname']/parent::div/parent::div/following-sibling::div/child::div[2]/child::in
18        WebElement fullName= driver.findElement(By.xpath("//input[@name='04lastname']/parent::div/parent::div/following-sibling::div/child::input"));
19        fullName.sendKeys("Jacob John");
20    }
21 }

```

Figure 18:Selenium: XPath

Alerts & PopUps:

- A small message box that appears on screen to give user some information
- It notifies the user on some error / warnings or asks for some permission to perform certain task
- Simple Alert - displays some information or warning
- Confirmation Alert - will ask for any permission
- Prompt Alert - will ask for some input from user; key in any value & close the alert
- Popups / Ads - Part of web page, content that gives some information & closed by script


```
Get Started package ex2_SeleniumExamples; Untitled-1 Extension: Extension Pack for Java
1 package ex2_SeleniumExamples;
2 import java.util.concurrent.TimeUnit;
3 import org.openqa.selenium.Alert;
4 import org.openqa.selenium.By;
5 import org.openqa.selenium.WebDriver;
6 import org.openqa.selenium.WebElement;
7 import org.openqa.selenium.chrome.ChromeDriver;
8
9 public class HandleAlerts {
10
11 public static void main(String[] args) throws InterruptedException {
12 System.setProperty("webdriver.chrome.driver", "E:\\Selenium\\Programs\\Team_INTSD002_03_Selenium\\Browsers\\chromedriver.exe");
13 WebDriver driver = new ChromeDriver();
14 driver.manage().window().maximize();
15 driver.get("file:///E:/JavaScript/Example1/input.html");
16 driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
17 WebElement name = driver.findElement(By.name("name"));
18 name.sendKeys("Jacob");
19 WebElement print = driver.findElement(By.xpath("//input[@value]"));
20 print.click();
21 Thread.sleep(3000);
22 Alert alert = driver.switchTo().alert();
23 System.out.println("Name in alert .. " + alert.getText());
24 alert.accept();
25 }
26 }
```

Figure 19:Selenium: Alerts and Pop Ups

Waits:

Most applications are developed with some dynamic contents & appear in the web page at a desired time. When a web page is loaded by browser, the elements may load at different time intervals.

- **Implicit Wait**
 - Used to tell the web driver to wait for certain amount of time, so all the elements of web page are loaded
 - Otherwise it would throw “NoSuchElementException”
- **Explicit Wait**
 - Used to tell the web driver to wait for certain conditions or maximum time on an element, so that particular element is loaded
 - Otherwise it will thro “ElementNotVisibleException”
 - Intelligent wait, since it is applied to specific elements
 - Gives lots of options than implicit wait for dynamic contents

```
Get Started package ex2_SeleniumExamples; Untitled-1 Extension: Extension Pack for Java
1 package ex2_SeleniumExamples;
2 import java.util.concurrent.TimeUnit;
3 import org.openqa.selenium.Alert;
4 import org.openqa.selenium.By;
5 import org.openqa.selenium.WebDriver;
6 import org.openqa.selenium.WebElement;
7 import org.openqa.selenium.chrome.ChromeDriver;
8
9 public class HandleAlerts {
10
11     public static void main(String[] args) throws InterruptedException {
12         System.setProperty("webdriver.chrome.driver", "E:\\Selenium\\Programs\\Team_INTSD002_03_Selenium\\Browsers\\chromedriver.exe");
13         WebDriver driver = new ChromeDriver();
14         driver.manage().window().maximize();
15         driver.get("file:///E:/JavaScript/Example1/input.html");
16         driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
17         WebElement name = driver.findElement(By.name("name"));
18         name.sendKeys("Jacob");
19         WebElement print = driver.findElement(By.xpath("//input[@value]"));
20         print.click();
21         Thread.sleep(3000);
22         Alert alert = driver.switchTo().alert();
23         System.out.println("Name in alert .. " + alert.getText());
24         alert.accept();
25     }
26 }
```

Figure 20:Selenium: Waits

Java Script Executor:

A point of interaction that gives a component to execute Javascript through Selenium WebDriver. At the point when Selenium webdriver finders (like XPath, name, id, and so forth) don't find a component, the JS executor performs the desired task.

There are two methods how this is done:

- **ExecuteScript**
 - Runs in the body of anonymous function
 - Allows to pass Boolean, long, string, list, webelements as arguments
 - Blocks any other further action by browser
 - Any code in the script will be executed by browser & not server
- **ExecuteAsyncScript**
 - Help render the page more quickly
 - Does not block any other further action by browser
 - Will help to improve the performance of the test

```
Get Started package ex2_SeleniumExamples; Untitled-1 Extension: Extension Pack for Java
1 package ex2_SeleniumExamples;
2
3 import java.util.concurrent.TimeUnit;
4 import org.openqa.selenium.By;
5 import org.openqa.selenium.JavascriptExecutor;
6 import org.openqa.selenium.WebDriver;
7 import org.openqa.selenium.WebElement;
8 import org.openqa.selenium.chrome.ChromeDriver;
9
10 public class JSExecutorSample {
11
12     public static void main(String[] args) {
13         System.setProperty("webdriver.chrome.driver", "E:\\Selenium\\Programs\\Team_INTSD002_03_Selenium\\Browsers\\chromedriver.exe");
14         WebDriver driver = new ChromeDriver();
15         driver.manage().window().maximize();
16         driver.get("https://www.facebook.com/");
17         //driver.get("https://www.robofarm.com/filling-test-all-fields");
18         driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
19         WebElement email = driver.findElement(By.id("email"));
20         email.sendKeys("abc@gmail.com");
21         WebElement pwd = driver.findElement(By.id("pass"));
22         pwd.sendKeys("asdasd");
23         WebElement login = driver.findElement(By.name("login"));
24         JavascriptExecutor js = (JavascriptExecutor)driver;
25         js.executeScript("arguments[0].click()", login);
26         js.executeScript("alert('Login successful')");
27         // JavascriptExecutor js = (JavascriptExecutor)driver;
28         // js.executeScript("window.scrollTo(0,500)");
29         // WebElement comment = driver.findElement(By.name("72_commnt"));
30         // comment.sendKeys("sdfdsdfsdf");
31     }
32 }
```

Figure 21: Selenium: JS Executor

```
Get Started package ex2_SeleniumExamples; Untitled-1 Extension: Extension Pack for Java
1 package ex2_SeleniumExamples;
2 import java.util.Set;
3 import java.util.concurrent.TimeUnit;
4 import org.openqa.selenium.By;
5 import org.openqa.selenium.WebDriver;
6 import org.openqa.selenium.WebElement;
7 import org.openqa.selenium.chrome.ChromeDriver;
8
9 public class HandleTabs {
10
11 public static void main(String[] args) throws InterruptedException {
12 System.setProperty("webdriver.chrome.driver", "E:\\Selenium\\Programs\\Team_INTSD002_03_Selenium\\Browsers\\chromedriver.exe");
13 WebDriver driver = new ChromeDriver();
14 driver.manage().window().maximize();
15 driver.get("https://www.nopcommerce.com/en/showcase");
16 driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
17 System.out.println("Current Window title .. " + driver.getTitle());
18 System.out.println("Focus is on current window .. " + driver.getWindowHandle());
19 String currWindow = driver.getWindowHandle();
20 WebElement image = driver.findElement(By.xpath("//img[@title='Volvo Cars Collection']"));
21 image.click();
22 Set<String> setAllWindows = driver.getWindowHandles();
23 System.out.println("Total windows opened .. " + setAllWindows.size());
24 int count=1;
25 for (String window : setAllWindows){
26 System.out.println("Window (" + count+ ") is " + window);
27 count++;
28 if (currWindow.equals(window)){
29 System.out.println("you are in first window.." + driver.getCurrentUrl());
30 }
31 else
32 {
33 driver.switchTo().window(window);
34 System.out.println("New Window title .. " + driver.getTitle());
35 System.out.println("Focus is on new window .. " + driver.getWindowHandle());
36 Thread.sleep(3000);
37 driver.close();
38 }
39 }
40 driver.switchTo().window(currWindow);
41 System.out.println("After switch back, First Window title .. " + driver.getTitle());
42 System.out.println("After switch back, Focus is on First window .. " + driver.getWindowHandle());
43 }
44 }
```

Figure 21: Selenium: handling multiple tabs of browser

Chapter 4 SQL Server

4.1 Introduction to RDBMS:

RDBMS stands for Relational Database Management System. A relational database is a database category which will store data points and is able to provide access to data points which are related to each other. Relational databases are built on the concept of relational model which is an intuitive and straightforward method of tabular representation of data. In a relational database there is a unique ID called the key for each row in the. The table's columns include data attributes, and each record typically has a value for each attribute, making it simple to construct links between data points.

4.2 Structured Query Language (SQL) :

SQL is a computer language for retrieving and managing data in relational databases. Structured Query Language stands for Structured Query Language. SQL enables us to conduct operations on the database's records. Update records, delete records, create and edit database tables, insert records views, and so on are examples of such operations. SQL is the standard language which is used for Relational Database Systems. MySQL, MS Access, Oracle, Sybase, Informix, Postgres and SQL Server use SQL as the standard database language for all the Relational Database Management Systems (RDBMS).

4.2.1 Applications of SQL

SQL is the most popular query language that is used to manage databases.

- It facilitates us to access data in the relational database management systems.
- It helps us to describe the data.
- It provides us with tools to define the data in a database and manipulate that data.
- It provides us with tools to embed within other languages using SQL modules, libraries & pre-compilers.
- It provides us with tools to create and drop databases and tables.

- It provides us with tools to create views, stored procedures, functions in a database.
- It provides us with tools to set permissions on tables, procedures and views.

When we need to run a SQL command on a relational database management system, the system automatically determines the optimal approach to complete our request, and the SQL engine decides how to interpret that command.

4.2.2 SQL Components :

- Query Dispatcher
- Optimization Engines
- Classic Query Engine
- SQL Query Engine, etc.

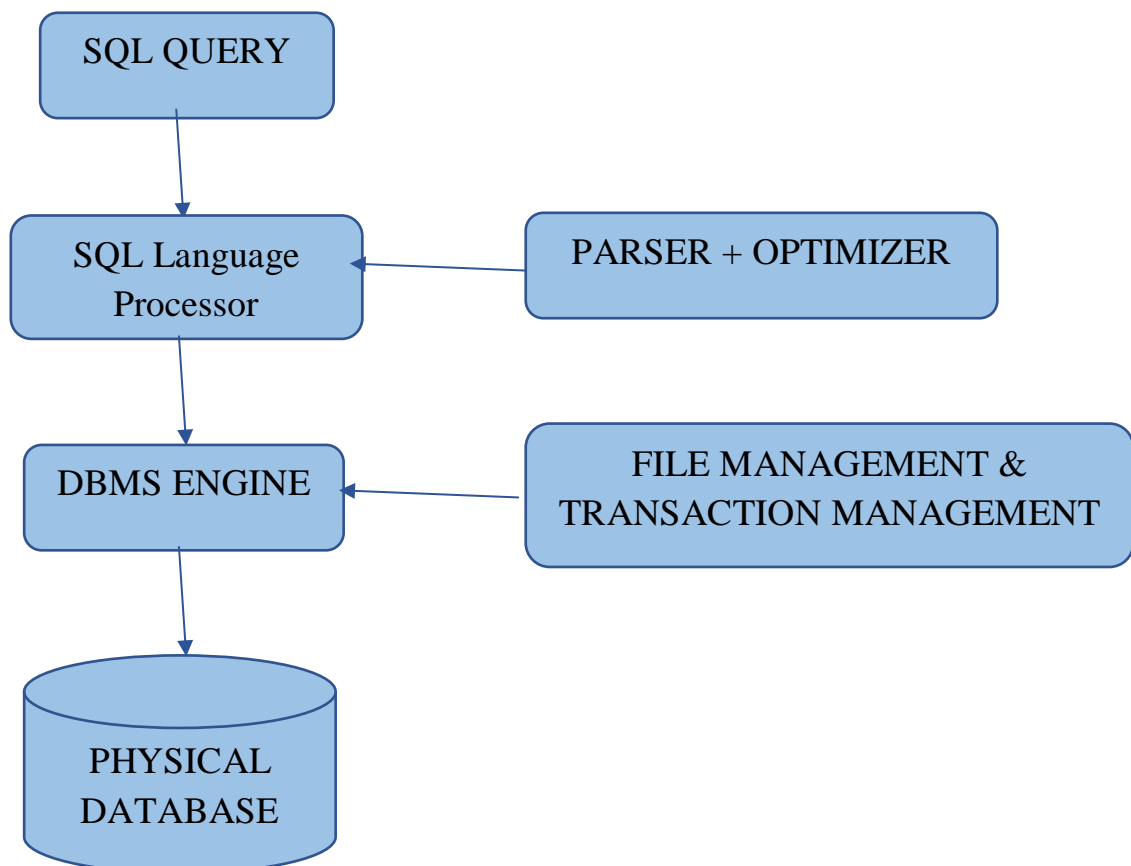


Figure 35: SQL Components Processing

4.2.3 SQL Architecture

SQL Server is a product of Microsoft, which obeys the principles of relational database management system (RDBMS). The basic functionality of the server is to store and thereby perform the retrieval operation on the data. The SQL server is completely based on a standard programming language i.e, SQL particularly used for working with relational databases. For over a span of two decades, SQL Server served the technical world for Windows. And soon in 2016, Microsoft released it for Linux. SQL Server Architecture

SQL Server consists of two main components:

- Database Engine
- SQLOS

Database Engine:

Database Engine is the core component of the SQL Server. The Database Engine consists of two things. A relational engine which is used to process queries and a storage engine which is used to manage database files, indexes, pages, etc.

- Relational Engine:
The Relational Engine consists of the components which decide the best way to complete a query. The query processor is another name for the relational engine. Based on the input query, the relational engine retrieves data from the storage engine and processes the results. Query processing, memory management and distributed query processing are all tasks performed by the relational engine.
- Storage Engine:
The storage engine is responsible for data storage and retrieval from storage systems like discs and SAN.

SQLOS:

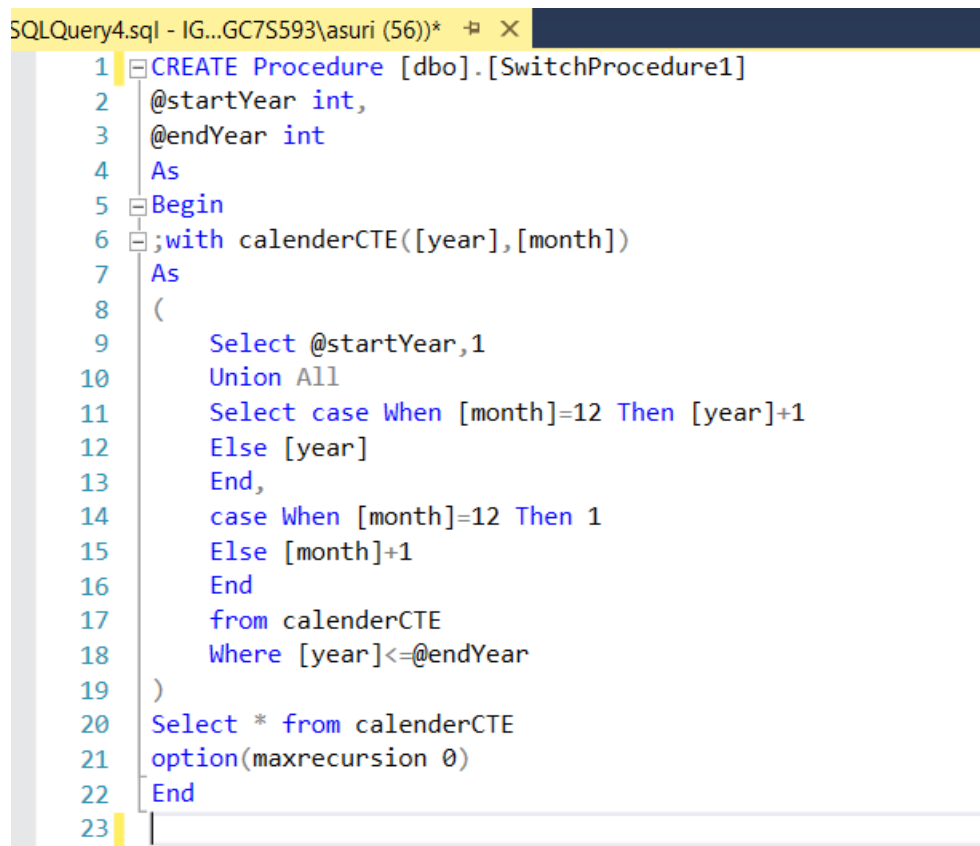
The SQL Server Operating System, or SQLOS, sits beneath the relational engine and storage engine. Many operating system services, including memory and I/O management, are provided by SQLOS. Exception handling and synchronization services are among the other services available

4.3 Stored Procedures:

SQL Server provides Stored Procedures with the motive of grouping multiple SQL statements in logical units. This procedure is a named object when it comes to the storage in SQL Server Database Server. On calling it at first, SQL Server creates a plan that handles the execution, called the execution plan and storage is done in the cache. On further executions, SQL Server reuses the plan for a fast and reliable execution.

The main advantages of stored procedure:

1. **Better Performance:** The stored procedure calls are quick and efficient. The reason being stored procedures are compiled once and then are stored in executable form thus giving quick response.
2. **Higher Productivity:** As we are reusing the same piece of code again and again so productivity is higher.
3. **Scalability:** By isolating application processing on the server, SP increases scalability.

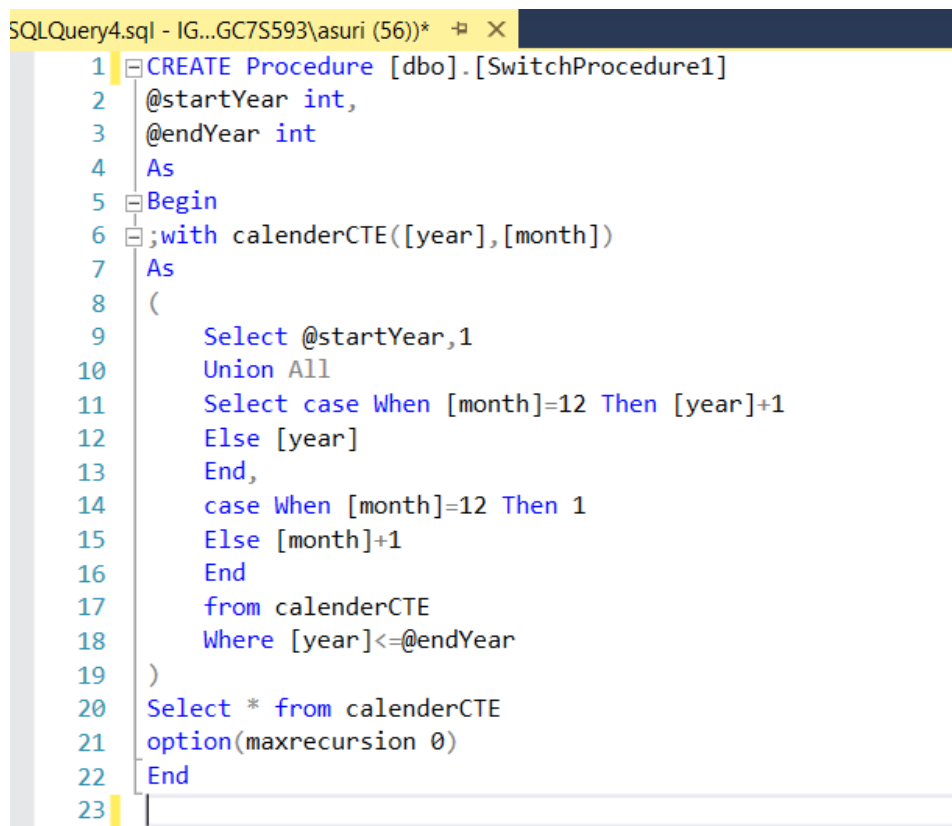


```
SQLQuery4.sql - IG...GC7S593\asuri (56))*  X
1 CREATE Procedure [dbo].[SwitchProcedure1]
2   @startYear int,
3   @endYear int
4   As
5 Begin
6   ;with calenderCTE([year],[month])
7   As
8   (
9     Select @startYear,1
10    Union All
11    Select case When [month]=12 Then [year]+1
12     Else [year]
13    End,
14    case When [month]=12 Then 1
15     Else [month]+1
16    End
17    from calenderCTE
18    Where [year]<=@endYear
19  )
20  Select * from calenderCTE
21  option(maxrecursion 0)
22  End
23
```

Figure 36: Stored Procedure

4.4 Common Table Expressions (CTE):

CTE stands for common table expression. A CTE helps us to create a temporary named result set which has temporary availability in the execution scope of a statement such as SELECT, INSERT, UPDATE, DELETE, or MERGE. This clause can be used in multiple situations like it can be used in a CREATE VIEW statement as part of its defining SELECT statement. A common table expression can even refer to itself. This is known as a recursive CTE.

The image shows a screenshot of a SQL Server query window. The title bar reads "SQLQuery4.sql - IG...GC7S593\asuri (56)*". The query text is as follows:

```
1 CREATE Procedure [dbo].[SwitchProcedure1]
2   @startYear int,
3   @endYear int
4   As
5   Begin
6     ;with calenderCTE([year],[month])
7     As
8     (
9       Select @startYear,1
10      Union All
11      Select case When [month]=12 Then [year]+1
12      Else [year]
13      End,
14      case When [month]=12 Then 1
15      Else [month]+1
16      End
17      from calenderCTE
18      Where [year]<=@endYear
19    )
20    Select * from calenderCTE
21    option(maxrecursion 0)
22  End
23
```

Figure 37: CTE

4.4.1 Recursive CTE:

A CTE that references itself is known as a recursive common table expression. By referring to itself, the CTE is continually executed, returning subsets of data until the entire result set is returned. When we need to query hierarchical data, such as organization charts where one employee reports to a manager or multi-level bill of materials where a product is made

up of many components, each of which is made up of many more components, a recursive CTE comes in handy.

```

SQLQuery6.sql - IG...GC7S593\asuri (58)*  X
1 ALTER Procedure [dbo].[finalCTE]
2   @startYear int,
3   @endYear int
4   AS
5   BEGIN
6       DECLARE @year int, @month int
7       SET @year = @startYear
8       SET @month = 1
9       ; with CTE (year1)
10      AS
11      (
12          SELECT @year
13          UNION ALL
14          SELECT year1+1 FROM CTE
15          WHERE year1 < @endYear
16      ),
17      CTE2 (month1)
18      AS
19      (
20          SELECT @month
21          UNION ALL
22          SELECT month1+1 FROM CTE2
23          WHERE month1 < 12
24      )
25      SELECT * FROM CTE2 CROSS JOIN CTE WHERE
26      NOT(NOT((year1 % 4 = 0 AND year1 % 100 <> 0) OR year1 % 400 = 0)and month1=2)
27      OPTION(maxrecursion 0)
28  END

```

Figure 38: Recursive CTE

4.5 PIVOT:

In SQL Server, the PIVOT operator perform rotation on a table-valued expression. It is used to split a single column's unique values across numerous columns and to aggregate any residual column values in the final result.

```

19 SELECT Teacher, Chef FROM #newTable
20 PIVOT
21 (
22     max(nameData)
23     For professionName
24     IN ([Teacher], [Chef])
25 ) AS PivotTable

```

Figure 39: Pivot

4.6 DYNAMIC SQL:

Dynamic SQL is a programming approach that allows us to create SQL statements on the fly. Because the finalized language of the SQL statements may be unknown at compilation, it allows us to build more generalized and flexible SQL statements. Dynamic SQL can be used to create a stored procedure that executes a query on data against an unknown table at runtime.

4.7 SQL Server Management Studio:

SQL Server Management Studio (SSMS) is a software tool for configuring, controlling, and administering all Microsoft SQL Server components. It was first released with Microsoft SQL Server 2005. It replaces the Enterprise Manager in SQL 2000 or earlier. The tool offers both script editors and graphical interfaces for working with the server's objects and functionalities.

SSMS is a SQL Server management tool that can be used to construct queries and administer databases and data warehouses through a personal computer or the cloud, independent of your location. SQL Server Management Studio (SSMS) is basically an integrated environment with tools for configuring, monitoring, and administering SQL Server machines and databases.

The Object Explorer is a key element of SSMS, allowing the user to view, select, and act on any of the server's objects

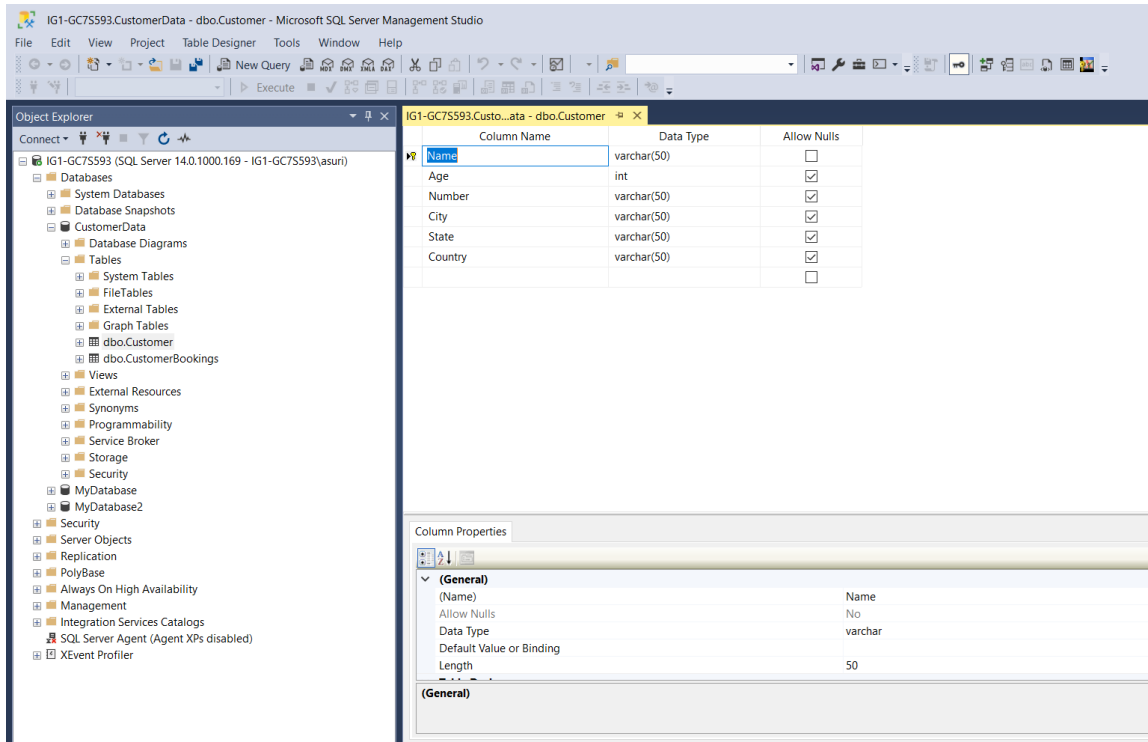


Figure 40: SQL Server Management Studio Environment

4.8 Object Relational Model (ORM):

An ORM stands for object relational mapping. This just means that we map SQL objects which are relational and you have foreign keys and primary keys to actual objects and code so that way we can more easily both read and view respond in objects and also manipulate them so that way you can set properties and say like save changes instead of needing to think about how to do an update statement. ORM allows us to write in the language you are already using. SQL is a very powerful language, but at the same time it is difficult to write.

Need to use ORM

Assume we wish to create an application to handle a college's students. To accomplish so, we may need to construct classes like Student, Department, Address, and so on. These classes were formally known as Domain classes.

We have to write a lot of data access code to do the CRUD activities, i.e. store and retrieve the Student, Department, and Address data from the underlying database tables, if we don't use an ORM framework like Entity Framework or EF Core.

To conduct CRUD actions, such as read, insert, update, or delete from a database table, we typically need to write custom code in our application to generate the necessary SQL

statements that the underlying database can understand. When we wish to read data from a database into our application, we must build some custom code to translate the database data to our model classes, such as Student, Department, and Address. As a developer, this is a task that we perform in practically every application.

If we submit the necessary information to the ORM Framework, such as Entity Framework or EF Core, it can accomplish all of the above for us and save us a lot of time. Between our application code and the database is the ORM Framework. It eliminates most of the custom data-access code that we would normally create without an ORM.

4.9 Entity Framework Core:

Entity Framework (EF) Core is a data access ORM (Object-Relational Mapper) framework for .NET. It is an extendable, lightweight, Open Source, and cross-platform version of Entity Framework data access technology that was released alongside .NET Core. It runs on a variety of operating systems, including Windows, Mac, and Linux. It is an ADO.NET extension that provides developers with an automated technique for accessing and storing data in databases.

4.9.1 Entity Framework Core has two approaches:

Code-First Approach :

First, we must develop our application domain classes such as Student, Branch, Address, and a specific class that derives from the Entity Framework DbContext class in the EF Core Code First Approach. The EF Core then generates the database and related tables based on the application domain classes and the DbContext class. The EF Core API generates the database and tables using migration based on default conventions and setup in the code-first approach. You can also change the default conventions for creating the database and its related tables if you like.

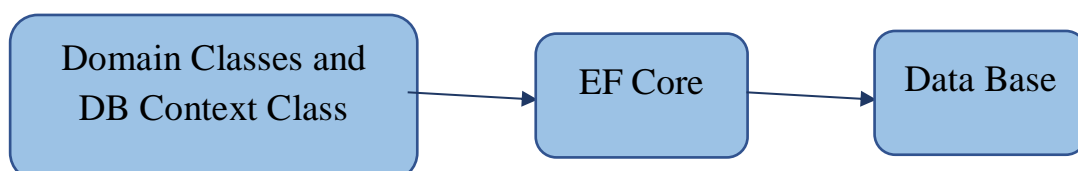


Figure 41: Model of Code First Approach

EF Core Database First Approach:

If database and database tables already exist then you should utilise the EF Core Database First Approach. EF Core constructs the DbContext and Domain classes based on the existing database schema using EF Core Command in the database-first method.

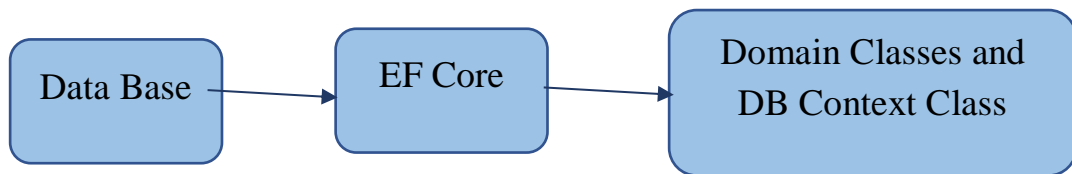


Figure 42: Model Of Database First Approach

Column Name	Data Type	Allow Nulls
Name	varchar(50)	<input type="checkbox"/>
Age	int	<input checked="" type="checkbox"/>
Number	varchar(50)	<input checked="" type="checkbox"/>
City	varchar(50)	<input checked="" type="checkbox"/>
State	varchar(50)	<input checked="" type="checkbox"/>
Country	varchar(50)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Figure 43: Design of Customer Table

```

using System;
using System.Collections.Generic;

namespace NewCustomerData.Models
{
    10 references
    public partial class Customer
    {
        4 references
        public string Name { get; set; }
        0 references
        public int? Age { get; set; }
        1 reference
        public string Number { get; set; }
        1 reference
        public string City { get; set; }
        1 reference
        public string State { get; set; }
        2 references
        public string Country { get; set; }
    }
}
  
```

Figure 44: Customer Class automatically generated by EF Core

4.10 EF Core Database Providers:

Because of database providers, the EF Core can support both relational and non-relational databases. NuGet packages are available for the database providers. Between the EF Core and the database it supports, the Database Provider sits.

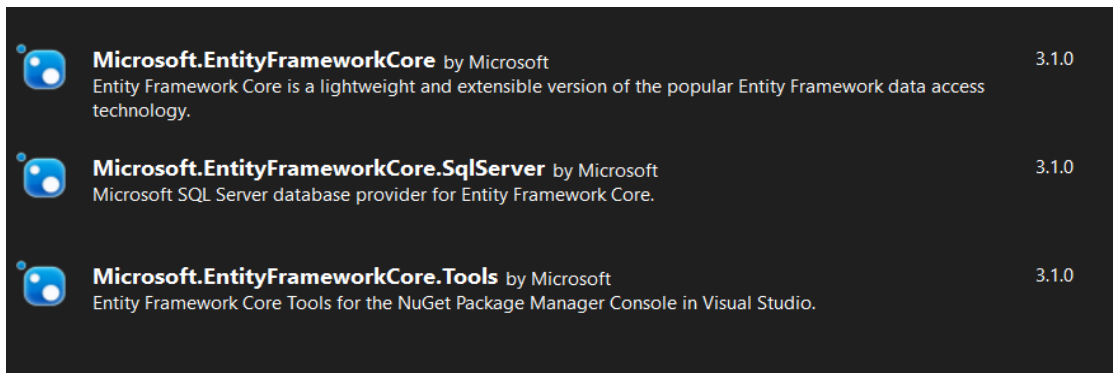


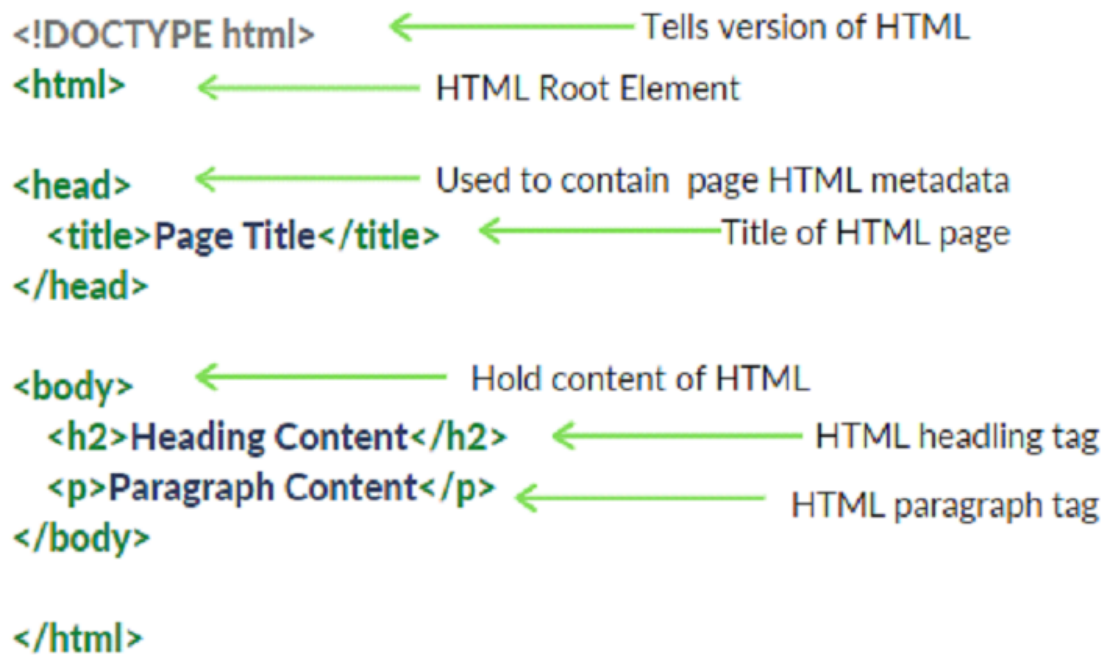
Figure 45: EF Core Dependencies

CHAPTER 5 : FRONT-END

5.1 HTML

HTML stands for HyperText Markup Language. It's a markup language that's used to create web pages. HTML is a markup language that combines hypertext with markup. The link between web pages is defined as hypertext. The text document within the tag that defines the structure of web pages is defined using a markup language. This language is used to annotate (add notes to) text so that a computer can understand it and manipulate it appropriately. Human-readable markup languages include HTML and others. Tags are used in the language to specify what type of text processing is required.

HTML is a markup language that the browser uses to transform text, pictures, and other content so that it can be displayed in the desired format. Tim Berners-Lee invented HTML in 1991. HTML 1.0 was the first version, but HTML 2.0 was the first standard version, released in 1995.



HTML Page Structure

Figure 46: HTML Page Structure

5.2 CSS- Cascading Style Sheets

CSS (Cascading Style Sheets) is a stylesheet language for describing the presentation of an HTML or XML document (including XML dialects such as SVG, MathML or XHTML). CSS describes how elements should look on a screen, on paper, in speech, or in other media.

CSS is a basic language for the open web that is supported by all major browsers. Separate parts of the CSS specification were previously developed in parallel, allowing for the versioning of the most recent suggestions.

CSS is not a programming language like HTML. It's also not a markup language. CSS is basically a style sheet language. CSS is used to style HTML components selectively.

Flex layout :

Flex Layout is included to give the component a neat appearance. The Flexbox is used to specify the children of a component layout. We can archive the correct layout by using the `flexDirection`, `justifyContent`, and `alignItems` attributes.

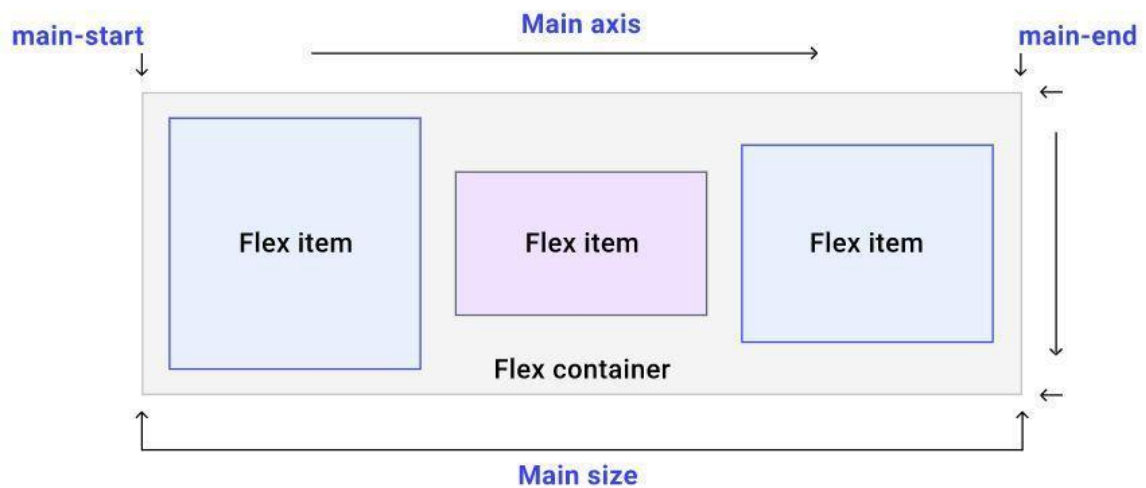


Figure 47: FlexBox Structure

5.3 Javascript

JavaScript is a lightweight, cross-platform object-based scripting language. JavaScript is a translated language, not a compiled language. The JavaScript Translator (which is built within the browser) is in charge of translating JavaScript code for web browsers.

JavaScript (js) is a lightweight object-oriented programming language that is used to script web pages by various websites. When applied to an HTML document, it is an interpreted, full-featured programming language that enables dynamic interactivity on websites. It was first released in 1995 to allow users to add programmes to web pages in the Netscape Navigator browser. Since then, all other graphical web browsers have embraced it. Users can utilize JavaScript to create modern web applications that interact immediately without having to reload the page every time. Js is used on a typical website to promote interactivity and simplicity.

There is no connection between JavaScript and the Java programming language. When Java began gaining prominence in the market, the name was suggested and provided. Databases like CouchDB and MongoDB use JavaScript as their scripting and query language, in addition to web browsers.

- **Synchronous JavaScript:** Synchronous refers to the execution of each statement of code in a sequential order. So, in essence, a statement must wait for the execution of the previous statement.
- **Asynchronous JavaScript:** Asynchronous code permits the program to be executed immediately, whereas synchronous code prevents the remaining code from being executed until the current one is completed. This may not appear to be a major issue, but when viewed in context, it can result in the User Interface being delayed.

5.3.1 AJAX

AJAX is an acronym that stands for Asynchronous JavaScript and XML. It is the use of the XMLHttpRequest object to communicate with servers in a nutshell. It can send and receive data in JSON, XML, HTML, and text files, among other formats. The most appealing characteristic of AJAX is its "asynchronous" nature, which allows it to talk with the server, exchange data, and update the website without having to refresh the page.

The two main functionalities of AJAX:

- Make requests to the server without reloading the page
- Receive and work with data from the server

5.4 Typescript

JavaScript was introduced as a client-side programming language. JavaScript has emerged as an emergent server-side technology as a result of the creation of Node.js. However, as JavaScript code develops in complexity, it becomes more difficult to maintain and reuse. Furthermore, JavaScript's failure to embrace Object Orientation, rigorous type checking, and compile-time error checks hinders it from prospering as a full-fledged server-side technology in the enterprise. To fill this void, TypeScript was introduced.

TypeScript is JavaScript for application-scale development.

TypeScript is a compiled, strongly typed, object-oriented language. TypeScript is a set of tools as well as a programming language. JavaScript is compiled to TypeScript, which is a typed superset of JavaScript. To put it another way, TypeScript is JavaScript with a few more features.

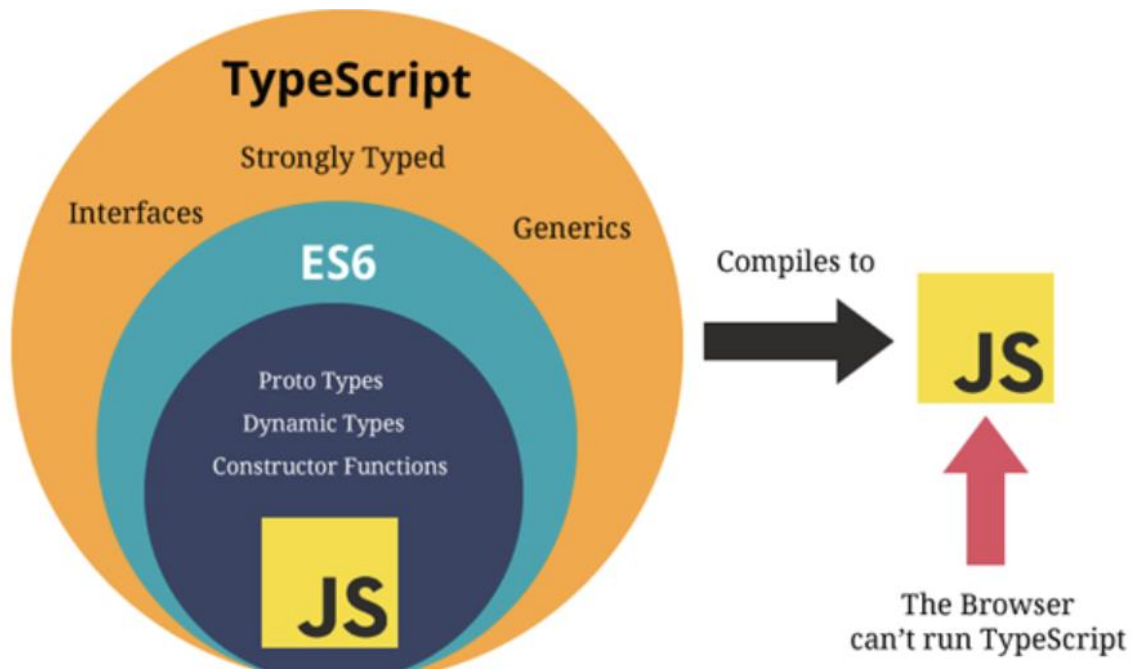


Figure 48: Compilation process of TypeScript

TypeScript is the outer domain of JavaScript. JavaScript is the beginning and the end of TypeScript. The essential building parts of your application are taken from JavaScript by TypeScript. To utilize TypeScript, you simply need to know JavaScript. For the purpose of execution, all TypeScript code is translated to its JavaScript equivalent.

JS libraries are supported by TypeScript. Any JavaScript code can consume compiled TypeScript. All existing JavaScript frameworks, tools, and libraries can be used with TypeScript-generated JavaScript.

JavaScript is TypeScript. JavaScript files can be converted to TypeScript just by renaming the .js file to .ts and compiled with other TypeScript files.

TypeScript is portable. TypeScript is cross-browser, device, and operating system compatible. It can run in any JavaScript-enabled environment. TypeScript does not require a separate virtual machine or a specific runtime environment to run, unlike its competitors.

Advantages of TypeScript

- TypeScript always highlights compilation mistakes during development (pre-compilation). Because JavaScript is an interpreted language, it is less likely to encounter runtime problems.
- Static/strong typing is supported in TypeScript. This means that type correctness can be verified during the compilation process. JavaScript does not support this feature.
- TypeScript is nothing more than JavaScript with some ES6 capabilities thrown in.

CHAPTER 6: CONCLUSION

This internship was indeed a pool of knowledge, not only have I gained knowledge in Full Stack Development but I have also learned about how development of any project takes place, how team works, how the work of each employee is tracked, how work is distributed between different team mates, what are the different stages of development, what are the technical problems that one faces in the development of any project, what all things are required before the development of any project, what the code base should be like and what norms need to be followed in the development. I have learned how to write code in such a manner that it's understood by every team member of my team. I have also gained some knowledge on being an active team member and leading my fellow team members to great ideas.

REFERENCES

- “Web APIs | MDN.” Web APIs | MDN, developer.mozilla.org, 14 Sept. 2021, <https://developer.mozilla.org/en-US/docs/Web/API>.
- MikeRayMSFT. “Tutorials for SQL Server - SQL Server.” Tutorials for SQL Server - SQL Server | Microsoft Docs, docs.microsoft.com, 17 Dec. 2021, <https://docs.microsoft.com/en-us/sql/sql-server/tutorials-for-sql-server-2016?view=sql-server-ver16>.
- “Object-Oriented Programming (C#) | Microsoft Docs.” Object-Oriented Programming (C#) | Microsoft Docs, docs.microsoft.com, 20 May 2022, <https://docs.microsoft.com/en-us/dotnet/csharp/fundamentals/tutorials/oop>.
- D.M. Rafi et al., "Benefits and limitations of automated software testing: Systematic literature review and practitioner survey", 7th International Workshop on Automation of Software Test (AST), pp. 36-42, 2012.