*A project report on*

# UTILITIES FOR BI PROJECTS

*Submitted in partial fulfillment for the award of the degree of*

## B. Tech-Information Technology

*by*

**AMANPREET TALUJA (18BIT0178)**



**SITE**

**May,2022**

# UTILITIES FOR BI PROJECTS

*Submitted in partial fulfillment for the award of the degree of*

## B. Tech-Information Technology

*by*

**AMANPREET TALUJA (18BIT0178)**



**SITE**

**May,2022**

**DECLARATION**

I hereby declare that the thesis entitled “Utilities For BI Projects” submitted by me, for the award of the degree of B.Tech-Information Technology, VIT is a record of bonafide work carried out by me under the supervision of Mr. Sumanth Pamulaparthy.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: Vellore

Date:5th May,2022 Signature of the Candidate

Internship completion certificate

**ABSTRACT**

Business intelligence (BI) refers to the procedural and technical infrastructure that collects, stores, and analyzes the data produced by a company’s activities.

BI is a broad term that encompasses data mining, process analysis, performance benchmarking, and descriptive analytics. BI parses all the data generated by a business and presents easy-to-digest reports, performance measures, and trends that inform management decisions.

The need for BI was derived from the concept that managers with inaccurate or incomplete information will tend, on average, to make worse decisions than if they had better information. Creators of financial models recognize this as “garbage in, garbage out.”

BI attempts to solve this problem by analyzing current data that is ideally presented on a dashboard of quick metrics designed to support better decisions.

**ACKNOWLEDGEMENT**

It is my pleasure to express with deep sense of gratitude to Mr.Sumanth Pamulaparthy,

Program Manager,Societe Generale, for his constant guidance, continual encouragement, understanding; more than all, he taught me patience in my endeavor. My association with him / her is not confined to academics only, but it is a great opportunity on my part of work with an intellectual and expert in the field of Transformations.

I would like to express my gratitude to <Chancellor>, <VPs>, <VC>, <PRO-VC>, and Dr. Sumathy S, SITE, for providing with an environment to work in and for his inspiration during the tenure of the course.

In jubilant mood I express ingeniously my whole-hearted thanks to <Program char- name>. <Program Chair and designation>, all teaching staff and members working as limbs of our university for their not-self-centered enthusiasm coupled with timely encouragements showered on me with zeal, which prompted the acquirement of the requisite knowledge to finalize my course study successfully. I would like to thank my parents for their support.

It is indeed a pleasure to thank my friends who persuaded and encouraged me to take up and complete this task. At last but not least, I express my gratitude and appreciation to all those who have helped me directly or indirectly toward the successful completion of this project.

Place: Vellore

Date:5th May,2022 Name of the student:

Amanpreet Taluja

## CONTENTS

[LIST OF FIGURES ix](#_TOC_250000)

LIST OF TABLES xi

LIST OF ACRONYMS xii

CHAPTER 1 BACKGROUND

* 1. BACKGROUND ON THE COMPANY 1
  2. BACKGROUND ON THE BUSINESS LINE 2

CHAPTER 2 INTRODUCTION

* 1. BI PROJECTS 10

**CHAPTER3**

**TECHNOLOGIES USED**

3.1 SPRING FRAMEWORK 10

3.2 REACT JS FRAMEWORK 10

**CHAPTER4**

**PROJECTS UNDERTAKEN**

4.1 STANDALONE UI FOR ONBOARDING TOOL……………………………………10

4.2 JAVA UTILITY TO ENTER DATA FROM JSON,EXCEL,CSV FILES TO POSTGRES DB.10

4.3 JAVA UTILITY TO ENTER DATA FROM APIs TO POSTGRES DB……………10

**CHAPTER5**

**MY LEARNINGS**

5.1 INDUSTRY EXPERIENCE …………………………………………………………10

**CHAPTER5**

**CONCLUSION & FUTURE WORK**

### **LIST OF FIGURES**

* 1. SOCIETE GENERALE GLOBAL SOLUTIONS CENTER 1
  2. TEAMSPIRIT 2
  3. INNOVATION 4
  4. RESPONSIBILITY 4
  5. COMMITMENT 4
  6. AREAS COVERED BY TREASURY 4
  7. BUSINESS INTELLIGENCE 4

3.1 SPRING FRAMEWORK 4

* 1. REACT JS 4

4.1 LEARNING MAP 4

* 1. PROP DRILLING 4

4.3 ASPOSE APACHE POI 4

**LIST OF EXAMPLE CODES**

4.1 MULTIPLE INSERT OPTIMIZATION 22

* 1. APOSTROPHE IN VARCHAR 48

**LIST OF ACRONYMS**

**Chapter 1**

**Background**

* 1. BACKGROUND ON THE COMPANY

1.1.1 HISTORY AND PRESENCE IN INDIA

Societe Generale was established in India in 1978. Today, Societe Generale operates through its local branches based in Mumbai and Delhi serving both:  
 -Indian corporates and financial institutions, helping them to expand their footprint outside of India   
 -Global clients of Societe Generale, from Europe, US, Middle-East, Africa and Asia-Pacific, to develop their operations in India



1.1.2 PRODUCTS AND SERVICES

In India, Societe Generale focuses on providing value-added solutions to its clients with its expertise in Global Markets and Structured Finance and also Cash Management and Trade Finance services. We are actively participating in India’s development to become an efficient project finance market notably on the renewable power side, with our association in International Solar Alliance.

1.1.3 GLOBAL SOLUTION CENTRE

Societe Generale Global Solution Centre (SGGSC), a 100% owned subsidiary of Societe Generale, provides services in the areas of Application Development and Maintenance, Infrastructure Management, Business Process Management, Research & Middle office and Knowledge Process Management to Societe Generale's business lines around the world.

SGGSC stands as a successful offshore development centre with 15 years of sustainable delivery experience, developing global best practices to promote the strategic objectives of Societe Generale. SGGSC has developed into a centre capable of delivering IT solutions as well as Business and Process consultancy.

Societe Generale Global Solution Centre’s journey as a service center for Societe Generale Group began in 2011 and had throughout the years a continuous development, enriched with projects that grew in variety and complexity.

For the first part of its journey, our company, initially named Societe Generale European Business Services, has been positioned as a valuable partner for the Group in its European operations, and this represented a vote of trust for our expertise and potential.

By listening to our clients’ needs and building on the expertise of our team, our company managed to create more and more services, by adding new business lines to its activity. Also, in the last couple of years, our service center worked together with our colleagues in India, as one team, towards a common goal: to create an undeniable positive global impact for Societe Generale Group. This expansion of our business impact has been reflected in our name as well, going from ‘European Business Services’ to ‘Global Solution Centre’.

Societe Generale Global Solution Centre Romania has grown substantially. From a single division specialized on financial services, in 2011, the company has transformed itself into a complex center offering a wide range of services and solutions: Finance & Accounting, Human Resources, IT, Know Your Customer or Compliance.

In 2012, HRCO was also opened to complement the company's service portfolio with Human Resources activities. In 2015, the launch of the IT Hub helped the Group expand its IT operations to Eastern Europe. Not only has it supported the increase of Societe Generale Global Solution Centre’s activity but has also helped to increase the portfolio of services offered.

Societe Generale is increasingly relying on its service centers, this being part of the Group's global transformation strategy. Therefore, our company decided to open a fourth line of business, delivering services to both Paris headquarters and other European entities. Our operations have developed and matured, so the services provided have split the activity into two new business lines: Know Your Customer and Compliance.

Now, Societe Generale Global Solution Centre is a service center for many Group entities, serving more than 35 countries for all Societe Generale's major business lines.

1.1.4 FOUR SHARED VALUES

Team spirit:



In a changing world, we support all our clients while demonstrating true team spirit. As a responsible and trusting partner, we provide attentiveness, agility and the complementary nature of our expertise. We work with our clients in the same way that we work with one another: by listening, building together, valuing contributions and being united in both our successes and our difficulties.

Innovation:



We strive to continually improve our client approach by leveraging technical innovation and working together to adapt our solutions and practices to the uses of the future. True to our entrepreneurial spirit, we are changing our ways of working by promoting sharing, experimentation and thinking outside the box. We learn from successes and failures alike.

Responsibility:



Our responsibility and our corporate ethics consist of meeting our clients’ various needs while safeguarding the long-term interests of all our stakeholders. Our responsibility can also be seen in our courage to be accountable for our actions and decisions, and to express our opinions transparently. We attach as much importance to how we achieve results as we do to the results themselves.

Commitment:



Our staff are fully committed to contributing to the success of our clients. We promote involvement and professional success for all. We nurture relationships based on trust and mutual respect, both inside and outside the Bank. Our commitment derives from the continued satisfaction of our clients in our skills and expertise.

Societe Generale encourages its staff to support the Group's not-for-profit partners in France and around the world. These volunteer programmes are drivers for social good that extend the Bank's practical reach. These drivers are essential for putting our values into practice.

1.1.5 CORPORATE PURPOSE

Engagements

Societe Generale was created over 150 years ago to promote the development of trade and industry. That was our original purpose, to be a bank that promoted progress.  We have always sought to support the long-term development of the global economy, through innovative solutions, which we have achieved by building a solid and sustainable bank with a pioneering spirit.

Our new purpose will guide future decisions, and reaffirm exactly it is that drives us when we come to work and serve our customers: the opportunity to continue playing a driving role in the world's transformations. Our "raison d'être" has been informed 85,000 contributions collected from staff across the world, in line with our history and the expectations of our stakeholders.

It is with this same long-term approach that we are engaging with the world as it is today: a world in which economic development goes hand-in-hand with environmental and social progress.

Building

Throughout its 150+ years of existence, Societe Generale has provided the financing needed for the world to transition from one development model to another. Societe Generale is a sustainable bank with a proven stability on which customers can depend. Building has always been the historical role of banks, in order to create the foundations for economic development.

Together

Team spirit is central to Societe Generale. Within the bank, we work together in order to offer each customer the tailored palette of expertise they require. We team up with our clients for the long term and are proud to be the trusted partner on which they rely to help them move the world forward. We establish coalitions with the aim of increasing the impact of positive changes.

With our clients

We assert our constant determination to be of service to our clients, whether businesses or individuals, by supporting them with their projects. Increasing the satisfaction of our clients is our priority.

A better and sustainable future

Through our support for our customers’ projects, we aim to help them build a better and sustainable future for themselves and for society. In the long run, we want to act in line with both individual and collective interests. Our vision is to play a leading role in the world’s positive transformations and contribute to a future that is more ecological, shows more respect for the planet and, promotes both economic, social and digital inclusion.

Solutions

Our approach focuses on finding value-added and suitable solutions to meet the needs of our customers and society, with a capacity to offer tailored solutions by drawing on our diverse expertise and experience in customer service.

Financial

As banker, we proudly embrace our profession and are mindful of the responsibility it comes with. Our business activities are needed now more than ever to finance, secure and support the projects of all those involved in building the world of tomorrow.

Responsible

We are committed to always taking an ethical approach with the willingness to preserve long-term trust and maintain relationships of loyalty. Furthemore, we wholeheartedly believe that the ability to make a positive impact on society and the environment will be central to the transformation of the banking sector. For these reasons, we see responsibility as an integral part of all the products and services we propose.

Innovative

We have a pioneering spirit and firmly believe that innovation is key to a more responsible world. Societe Generale helps to bring about new business models and creates unique solutions thanks to our capacity to combine different areas of expertise at the highest level.

* 1. BACKGROUND ON THE BUSINESS LINE

## WHAT IS TREASURY MANAGEMENT?

Treasuries are the custodians of cash in a business, they control this through

1) the amount held and 2) its liquidity. The two levers of this are through the sheer size of the balance sheet and the relative stickiness (liquidity) of assets and liabilities held. Their management of this enables the basic fundamentals of an organization: allowing teams to operate and conduct activities by ensuring that there is cash on hand, be it in the petty cash box or an opportunistic M&A raid.

In addition to enabling business-as-usual (BAU) activities, treasuries partake in the macro-financial direction of a company and oversee the execution of company-wide strategies. For example, if the board decides to buy a business or expand into new territories, Treasury will help to determine the fit of the company from a balance sheet perspective and find the cash (or issue stock) to purchase it ultimately.

By actively managing liquidity, treasuries ensure that businesses stay alive, save money, and can respond quickly to change.

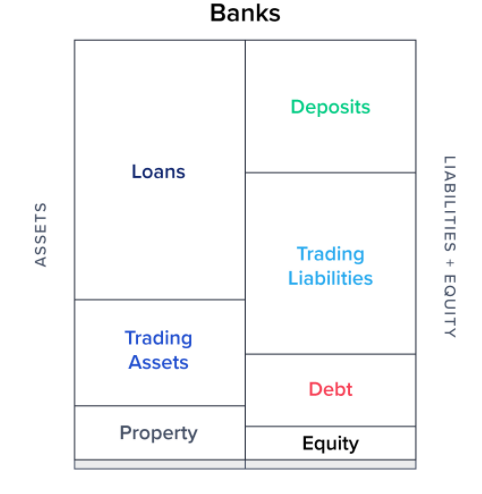
### 1.2.2 AREAS COVERED BY A TREASURY

The basics of treasury management can be distilled into five critical responsibilities.

ASSET LIABILITY MANAGEMENT (ALM)

ALM concerns the blend of assets and liabilities that sit on a balance sheet and the subsequent mismatches between tenor, currency, and interest rate (cost). Companies hold a range of instruments on balance sheets, which behave with varying characteristics. How they interact with each other and represent the overall position could be metaphorically described as being similar to the concept of Beta in portfolio management.

ALM is most relevant for treasury management in banks because their fundamental purpose is based upon the gearing dynamic of borrowing and lending money. The graphic below demonstrates generic balance sheet compositions for corporates and banks and, as you can see, banks are generally more leveraged through their increased use of liabilities relative to equity capital.



Because it’s generally cheaper to borrow short-term liabilities and invest in long-term assets, there is a natural tendency for companies to stretch this funding mismatch to a limit. This can all come crashing down during market flashpoints when credit dries up and liabilities become harder to roll. An ALM function monitors this liquidity horizon, prescribing limit buffers and advising on any changes that can be observed in advance.

Optimizing assets and liabilities in a proactive manner increases profitability and business opportunities. This is not just in the domain of banks; here are some examples of companies using treasury management to assist business:

Direct to consumer vendors (e.g., supermarkets, eCommerce) that have negative cash conversion cycles and offer consumer credit services.

Share repurchase schemes enacted at opportunistic periods.

Factoring receivables to gain a competitive advantage by winning new customers with attractive payment terms.

FUNDS TRANSFER PRICING (FTP)

Treasuries are mini-banks for their own companies (or banks) and must price up the liabilities on hand for use in everyday asset-generating activities. The FTP reflects the cost of liabilities and is charged to a business unit when it wishes to originate a new asset. Unlike the widely-known cost of debt figure, which can be represented as a standalone loan or benchmark bond yield, the FTP represents a fully-loaded cost. By that, I mean that it is the overall weighted average cost of all liabilities plus the internally shared costs of the business minus treasury profit.

Funds-transfer pricing is the process of costing a balance sheet and then setting the requisite prices for asset creators or liability gatherers to pay or earn for their respective tasks. Without this, there would be a free-for-all, with profitability and balance sheet structure left to its own devices.

TRADING AND HEDGING

The responsibilities of hedging company-wide interest rate and FX risk sits with the treasury function, who will use derivatives to balance the books. Depending on the sophistication of the business, these risk management strategies can range up from FX spot trades to long-term interest rate swaps.

For example, I worked at a bank with predominantly GBP-based liabilities, but with assets written in EUR. A sudden change in either currency would distort the risk, in terms of the proportions of the balance sheet and the relative profitability of deals. To counteract this, we would trade cross-currency swap derivatives to “crystalize” the asset positions into GBP to retain parity.

PORTFOLIO MANAGEMENT

Treasuries are financial asset managers for their company, investing spare cash that sits on the balance sheet to generate a return (and thus, lower FTP). This is often a very creative exercise that involves the search for yield, liquidity, and capital efficiency. Braeburn Capital, for example, is the asset management arm of Apple, a company that regularly has reserve treasury funds of over $200 billion!

INTEGRATION/PROJECTS

Overseeing all parts of the business and being agnostic towards any specific business line will usually put the treasury as a useful tool for integrating acquisitions into the company, or for spearheading IT transformation initiatives.

**Chapter 2**

**Introduction**

2.1 BI PROJECTS

2.1.1 WHAT IS BUSINESS INTELLIGENCE (BI)?

Business intelligence (BI) refers to the procedural and technical infrastructure that collects, stores, and analyzes the data produced by a company’s activities.

BI is a broad term that encompasses data mining, process analysis, performance benchmarking, and descriptive analytics. BI parses all the data generated by a business and presents easy-to-digest reports, performance measures, and trends that inform management decisions.

KEY TAKEAWAYS

* BI represents the technical infrastructure that collects, stores, and analyzes company data.
* BI parses data and produces reports and information that help managers to make better decisions.
* Software companies produce BI solutions for companies that wish to make better use of their data.
* BI tools and software come in a wide variety of forms such as spreadsheets, reporting/query software, data visualization software, data mining tools, and online analytical processing (OLAP).
* Self-service BI is an approach to analytics that allows individuals without a technical background to access and explore data.



## 2.1.2 UNDERSTANDING BUSINESS INTELLIGENCE (BI)

The need for BI was derived from the concept that managers with inaccurate or incomplete information will tend, on average, to make worse decisions than if they had better information. Creators of financial models recognize this as “garbage in, garbage out.”

BI attempts to solve this problem by analyzing current data that is ideally presented on a dashboard of quick metrics designed to support better decisions.

## 2.1.3 SPECIAL CONSIDERATIONS

To be useful, BI must seek to increase the accuracy, timeliness, and amount of data.

These requirements mean finding more ways to capture information that is not already being recorded, checking the information for errors, and structuring the information in a way that makes broad analysis possible.

In practice, however, companies have data that is unstructured or in diverse formats that do not make for easy collection and analysis. Software firms thus provide business intelligence solutions to optimize the information gleaned from data. These are enterprise-level software applications designed to unify a company’s data and analytics. Although software solutions continue to evolve and are becoming increasingly sophisticated, data scientists still need to manage the trade-offs between speed and the depth of reporting.

Some of the insights emerging from big data have companies scrambling to capture everything, but data analysts can usually filter out sources to find a selection of data points that can represent the health of a process or business area as a whole. This can reduce the need to capture and reformat everything for analysis, saving analytical time and increasing the reporting speed.

## 2.1.4 TYPES OF BI TOOLS AND SOFTWARE

BI tools and software come in a wide variety of forms. Let's take a quick look at some common types of BI solutions.

* **Spreadsheets:**Spreadsheets like Microsoft Excel and Google Docs are some of the most widely used BI tools.
* **Reporting software:** Reporting software is used to report, organize, filter, and display data.
* **Data visualization software:**Data visualization software translates datasets into easy-to-read, visually appealing graphical representations to quickly gain insights.
* **Data mining tools:**Data mining tools "mine" large amounts of data for patterns using things like artificial intelligence, machine learning, and statistics.
* **Online analytical processing (OLAP):**OLAP tools allow users to analyze datasets from a wide variety of angles based on different business perspectives.

## 2.1.5 BENEFITS OF BUSINESS INTELLIGENCE

There are many reasons why companies adopt BI. Many use it to support functions as diverse as hiring, compliance, production, and marketing. BI is a core business value; it is difficult to find a business area that does not benefit from better information to work with.

Some of the many benefits companies can experience after adopting BI into their business models include faster, more accurate reporting and analysis, improved data quality, better employee satisfaction, reduced costs, and increased revenues, and the ability to make better business decisions.

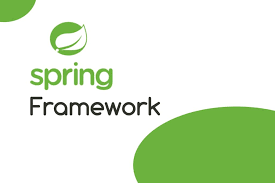
**Chapter 3**

**Technologies Used**

3.1 SPRING FRAMEWORK

3.1.1 INTRODUCTION

* Spring is the most popular application development framework for enterprise Java. Millions of developers around the world use Spring Framework to create high performing, easily testable, and reusable code.
* Spring framework is an open source Java platform. It was initially written by Rod Johnson and was first released under the Apache 2.0 license in June 2003.



* Spring is lightweight when it comes to size and transparency. The basic version of Spring framework is around 2MB.
* The core features of the Spring Framework can be used in developing any Java application, but there are extensions for building web applications on top of the Java EE platform. Spring framework targets to make J2EE development easier to use and promotes good programming practices by enabling a POJO-based programming model.

## 3.1.2 BENEFITS OF USING THE SPRING FRAMEWORK

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

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

## 3.1.3 ASPECT ORIENTED PROGRAMMING (AOP)

One of the key components of Spring is the Aspect Oriented Programming (AOP) framework. The functions that span multiple points of an application are called cross-cutting concerns and these cross-cutting concerns are conceptually separate from the application's business logic. There are various common good examples of aspects including logging, declarative transactions, security, caching, etc.

The key unit of modularity in OOP is the class, whereas in AOP the unit of modularity is the aspect. DI helps you decouple your application objects from each other, while AOP helps you decouple cross-cutting concerns from the objects that they affect.

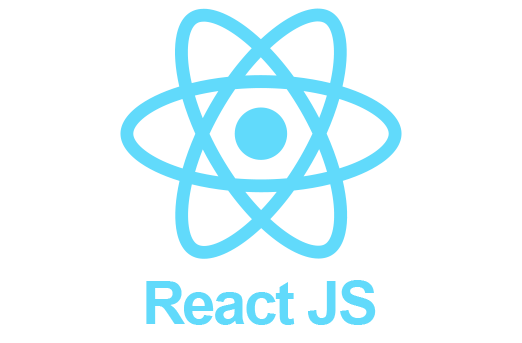
The AOP module of Spring Framework provides an aspect-oriented programming implementation allowing you to define method-interceptors and pointcuts to cleanly decouple code that implements functionality that should be separated. We will discuss more about Spring AOP concepts in a separate chapter.

3.2 REACT JS FRAMEWORK

3.2.1 INTRODUCTION

React (also known as React.js or ReactJS) is a [free and open-source](https://en.wikipedia.org/wiki/Free_and_open-source_software) [front-end](https://en.wikipedia.org/wiki/Front_end_and_back_end) [JavaScript library](https://en.wikipedia.org/wiki/JavaScript_library) for building [user interfaces](https://en.wikipedia.org/wiki/User_interfaces) based on UI components. It is maintained by [Meta](https://en.wikipedia.org/wiki/Meta_Platforms) (formerly Facebook) and a community of individual developers and companies. React can be used as a base in the development of [single-page](https://en.wikipedia.org/wiki/Single-page_application), mobile, or server-rendered applications with frameworks like [Next.js](https://en.wikipedia.org/wiki/Next.js).

However, React is only concerned with state management and rendering that state to the [DOM](https://en.wikipedia.org/wiki/Document_Object_Model), so creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality.



3.2.2 FEATURES

Declarative:

React adheres to the declarative programming paradigm. Developers design views for each state of an application, and React updates and renders components when data changes. This is in contrast with imperative programming.

Components:

React code is made of entities called components. These components are reusable and must be formed in the SRC folder following the Pascal Case as its naming convention (capitalize camelCase). Components can be rendered to a particular element in the DOM using the React DOM library. When rendering a component, one can pass the values between components through "props":

* + Functional components: Function components are declared with a function that then returns some JSX.
  + Class-based components: Class-based components are declared using ES6 classes. Where class components are all about the use of classes and the lifecycle methods, functional components have hooks to deal with state management and other problems which arise when writing code in React.

Virtual DOM:

Another notable feature is the use of a virtual Document Object Model, or virtual DOM. React creates an in-memory data-structure cache, computes the resulting differences, and then updates the browser's displayed DOM efficiently. This process is called reconciliation. This allows the programmer to write code as if the entire page is rendered on each change, while the React libraries only render subcomponents that actually change. This selective rendering provides a major performance boost. It saves the effort of recalculating the CSS style, layout for the page and rendering for the entire page.

Lifecycle methods:

Lifecycle methods for class-based components use a form of hooking that allows the execution of code at set points during a component's lifetime.

* shouldComponentUpdate: allows the developer to prevent unnecessary re-rendering of a component by returning false if a render is not required.
* componentDidMount: is called once the component has "mounted" (the component has been created in the user interface, often by associating it with a DOM node). This is commonly used to trigger data loading from a remote source via an API.
* componentWillUnmount: is called immediately before the component is torn down or "unmounted". This is commonly used to clear resource-demanding dependencies to the component that will not simply be removed with the unmounting of the component (e.g., removing any setInterval() instances that are related to the component, or an "eventListener" set on the "document" because of the presence of the component)
* render is the most important lifecycle method and the only required one in any component. It is usually called every time the component's state is updated, which should be reflected in the user interface.

JSX:

JSX, or JavaScript Syntax Extension, is an extension to the JavaScript language syntax. Similar in appearance to HTML, JSX provides a way to structure component rendering using syntax familiar to many developers. React components are typically written using JSX, although they do not have to be (components may also be written in pure JavaScript). JSX is similar to another extension syntax created by Facebook for PHP called XHP.

Architecture beyond HTML:

The basic architecture of React applies beyond rendering HTML in the browser. For example, Facebook has dynamic charts that render to <canvas> tags, and Netflix and PayPal use universal loading to render identical HTML on both the server and client.

React Hooks:

Hooks are functions that let developers "hook into" React state and lifecycle features from function components.Hooks do not work inside classes — they let you use React without classes.

React provides a few built-in hooks like useState, useContext, useReducer , useMemo and useEffect. Others are documented in the Hooks API Reference. useState and useEffect, which are the most commonly used, are for controlling state and side effects respectively.

## 3.2.3 BENEFITS OF REACT JS

The React JS offers tons of benefits. Let’s see the key benefits of React JS to understand why it stands out from other front-end development frameworks.

### Speed

The React basically allows developers to utilize individual parts of their application on both client-side and the server-side, which ultimately boosts the speed of the [development process](https://www.peerbits.com/process.html).

In simple terms, different developers can write individual parts and all changes made won’t cause the logic of the application.

### Flexibility

Compared to other frontend frameworks, the React code is easier to maintain and is flexible due to its modular structure. This flexibility, in turn, saves huge amount of time and cost to businesses.

### Performance

React JS was designed to provide high performance in mind. The core of the framework offers a virtual DOM program and server-side rendering, which makes complex apps run extremely fast.

### Usability

Deploying React is fairly easy to accomplish if you have some basic knowledge of JavaScript.

In fact, an expert JavaScript developer can easily learn all ins and outs of the React framework in a matter of a day or two.

### Reusable Components

One of the main benefits of using React JS is its potential to reuse components. It saves time for developers as they don’t have to write various codes for the same features. Furthermore, if any changes are made in any particular part, it will not affect other parts of the application.

**Chapter 4**

**Projects Undertaken**

4.1 STANDALONE UI FOR ONBOARDING TOOL

4.1.1 INTRODUCTION

Properly onboarding new employees’ costs time and resources but is well worth the investment. Why? A good onboarding experience can increase employee retention by up to 82% and productivity by 70%.

A seamless employee onboarding experience plays a vital role in improving employee performance and increasing retention. With employee onboarding software, HR managers can map the entire employee journey from recruiting to onboarding. This improves employee experience and boosts employee morale.

Bringing a new hire on successfully is a win-win. If they land well, there’s a much higher chance they’ll stay and perform.

Onboarding software creates a smooth, professional transition every time. It helps you build clear checklists and workflows to guide new hires through those all-important early days.

4.1.2 FRAMEWORKS USED:

* React JS
* Type Script
* SG Bootstrap
* Google Icons

4.1.3 PROCESS FOLLOWED:

The first step to creating the UI was to setup the SG standard White App for React based projects. It was mostly written in TypeScript and consisted of pre built functionalities such as:

* SG User Login
* SG Standard Header
* SG Standard Footer
* SG Standard Sidebar

which made it easier for me to focus on creating a Single Page Application with multiple screens.

The first screen I developed was a basic Login Form with various critical details which were to be used for Authentication, Authorization and for further use as well.

The screen was initially developed in a SocGen UI Development Software called a Playground where we can code as well as drag and drop pre formatted components onto the screen.The screens in playground are basically static and can only have HTML and CSS components.

After getting an approval on the static screen on Playground from my manager,my next task was to design this using React TypeScript in the SG White App.

Post this,my job was to handle Form button clicks and input validations which were specified to me by my manager according to Bussiness Logic.

I was ,very graciously,put in touch with an expert on Front End designing in the company,from whom I learnt a lot of designing principles and from whom I got a golden rule which may seem like common sense,but it is uncanny how many times one forgets using it:”Don’t overdo it,maintain consistency in styling over complexity”.

I understood the significance of this when my screen was reviewed by my Peer and the honest feedback I got was that:’it is too much to look at,please make it easy on the eyes’.

I learnt from this that standard and specific is better than complex and detailed when it comes to designing according to requirements.

The second screen I worked on was post a successful login,the user would be redirected to an organizational hierarchy relative to the user.The hierarchy was displayed in the form of a flow chart with important Point Of Contact’s mentioned for each department.

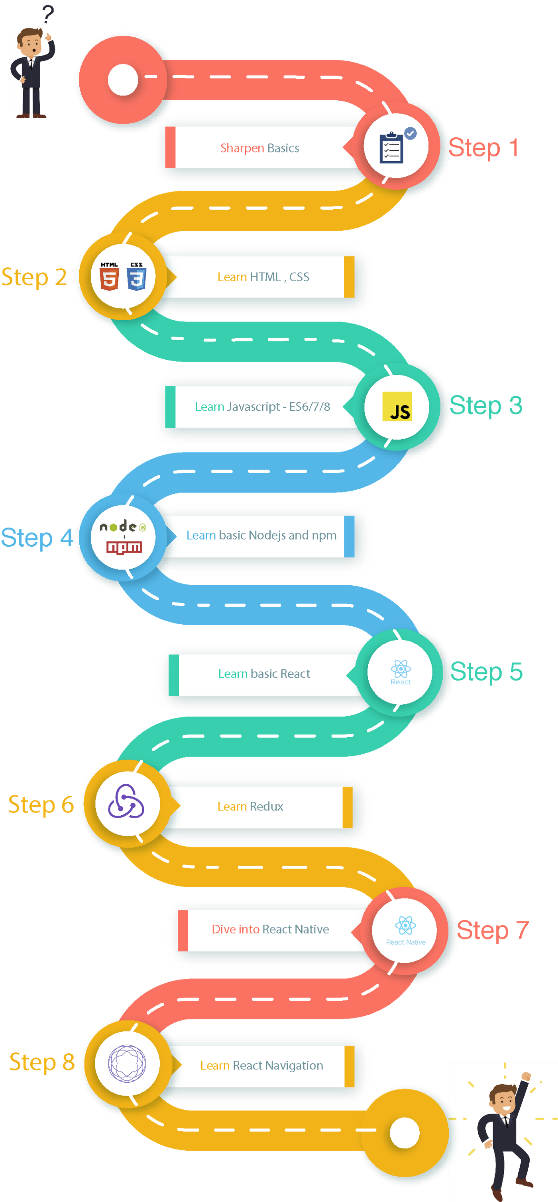
The third screen I worked on was the most important screen of the UI as it contained a LearningMap specific to each User.The LearningMap is a set of coiled Flow Charts which are required to be completed sequentially before you can access the next Learning Object.

This was done by splitting up the whole Learning Path into multiple domains such as:

* Access
* Your Organization
* Dev Essentials
* Functional Learning
* Fun Zone
* Key Contacts e.t.c.

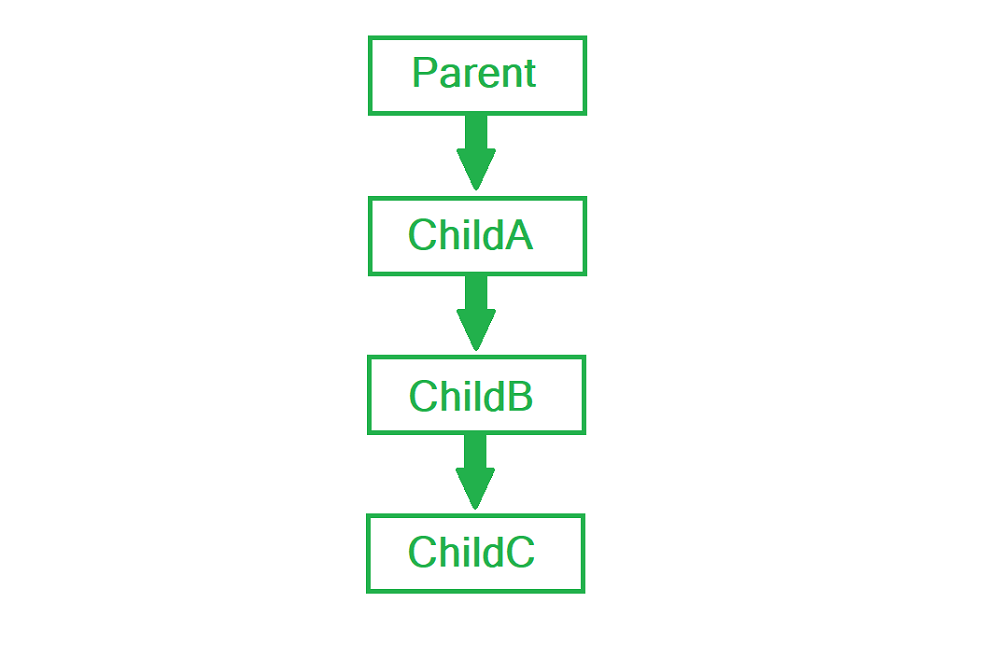
The main Challenge here was to make the Learning Map very very configurable,since it would be different from person to person depending on

1. Their Role in the company
2. The Team they would work with and
3. The Region of the World they were working from.



In the process of making the Learning Map very flexible and Configurable,I learnt the importance of Context Management and the difficulties faced when Drilling down props from one component to another through several layers.

Anyone who has worked in React would have faced this and if not then will face it definitely. Prop drilling is basically a situation when the same data is being sent at almost every level due to requirements in the final level. Here is a diagram to demonstrate it better. Data needed to be sent from Parent to ChildC. In this article different ways to do that are discussed.



To be fair, prop drilling isn’t all bad because the pattern provides you with a very specific way of passing data using a top-down approach. But sometimes, it’s just very annoying to specifically pass props down between components, especially when your React application is ten layers deep.

Imagine if you can just declare a global variable in any component and then retrieve it from any component. Without a clear pattern and one-way data flow, you’d have a very confusing data model and have a hard time tracking where some data is initialized, updated and used.

But React does acknowledge that it’s no fun to pass props down multiple components from the initial component. This is why React provides a way to simulate the nature of global variable. Enter the Context API.

4.2 JAVA UTILITY TO ENTER DATA FROM JSON,EXCEL,CSV FILES TO POSTGRES DB

4.2.1 INTRODUCTION

The main objective of the tool was to be able to develop a common utility which can extract data from various file based data sources in three particular formats ,i.e,

* CSV
* JSON
* XLS

and enter it into specific tables in a Postgres DB Cluster for the data to be processed and used further for visualizations**.**

4.2.2 WORK DONE

The main work for me was to be able to convert CSV into XLS as data from XLS can be easily read using RowIterators and CellIterators for each row.The most efficient way I found to do so was to open the csv file using an Apache Aspose workbook,setting the delimiter and saving the workbook with a .xlsx extension.On the performance side of things this conversion used to take about 200 seconds fro converting a csv containing 10lakh records with 28 columns each into a comma separated xlsx.



Once this was done the main performance issue was to be reading an xlsx which would typically have 10 lakh records all at once into a workbook.This came up as an issue because it used to take up more than 5Gigs of memory and an ample amount of overhead time to do so.

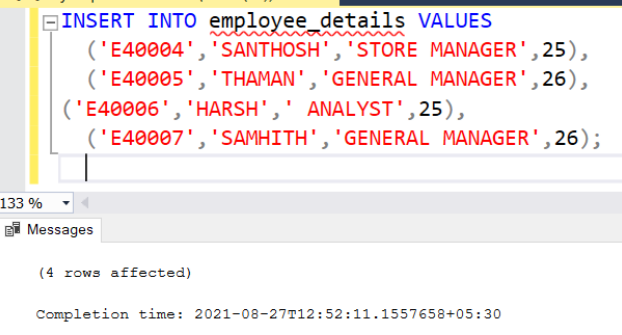
To prevent the overhead memory and time I decided to go with a buffering scheme called the StreamingReader which reads a set number of rows from a FileInputStream and for my Use Case I set it as 10,000 rows at a time.This greatly reduced overhead time and memory required to process the rows into the workbook.

The next step was to ignore an initial couple of rows as they act as headers and then for each subsequent row,get all cell values using a CellIterator enter them into a precompiled sql insert statement and once there were no more cells in the Row,execute the insert statement using an AutoWired RestTemplate.

This approach seemed to be working fine until we tested it with sample data,where the performance was basically insertion of 4 records per second and with company data sources containing lakhs and lakhs of records to be inserted daily,this performance metric would not cut it.

4.2.3 OPTIMIZATIONS MADE

What I learnt form this was that each time you passed an Insert statement,it took up an entire DB session to execute it.Following from the principles of a transaction statement,I tried to experiment with passing multiple records using a single insert statement as shown below:



This seemed to work extremely well and I started experimenting with the number of records that needed to be inserted in a single Insert statement.This testing was important since I had to balance out the read time taken by the utility to parse each row and the time taken by the Insert command to be executed.

After sufficient enough testing I found the optimal number of records to be 500.With this I could achieve about 1,200 rows inserted per second which was exponentially better than what I was able to achieve before.

In an attempt to try and do better than this,the next step I took was to implement multithreading.I had read a lot about it in a lot of programming classes but had never experienced its power before.I tried implementing it initially by creating individual runnables but they seemed to be running synchronouslyand therefore not actually using multiple threads at once.The next thing I tried,was creating an Executorservice consisting of a Cached Thread Pool.

What this implied was that as soon as an Insert statement was ready to be executed ,the utility would create a new thread and start working on readying a new insert statement by carrying out read functions on the next 500 rows in the workbook.This meant that the insertion time of one batch and the read time of the next batch of rows overlapped resulting in much lesser overhead per batch and me being able to push my performance from a 1,200 rows entered per second to a 2,300 rows being entered per second.

In an effort to make the tool more configurable we used the @Value annotation of SprinBoot to be able to fetch table name,file name and other such inputs from the User by fetching these values from the application.properties file.

The utility also has a feature of being able to create a table in the database by reading a configure.xlsx file placed alongside its executable.To create the table,we need to mention the column names and their respective datatypes in the first 2 columns of the excel sheet.Upon running the utility,the code goes throught the excel sheet and adds the configurable table name,column names and their datatypes to a pre configured sql query and then executes the simple create table statement using the JDBC template.

A copy of this configure file is also saved in a folder called schemas with the name as <table name>\_schema.xlsx for future reference of the user.

The next optimization we worked on was to be able to specify names of multiple files all at once for them to be added one after the other by running the utility just once.The only condition was that the names needed to be provided with a , delimiter so that the code would understand that it was to work with multiple files.We used the String Split function to implement this easily and save the file names in a String array.

4.2.4 ISSUES FACED

The biggest issue I seemed to be facing was insertion of a varchar containing a single quote in it,as it would wrongfully terminate the insert statement.The solution I found to this was to find the position of the single quote in the string and add a “\\” before it and an “e” before the string itself for Postgres to undertand that the single quote was not supposed to be recognized as an ending single quote but as a part of the string itself.

****

****

4.3 JAVA UTILITY TO ENTER DATA FROM APIs TO POSTGRES DB

4.3.1 INTRODUCTION

The need of the hour was to make a piece of Utility which would be able to fetch large amounts of data from an API and store it in a table in the database.

The three major compoinents to this task were:

1.Being able to generate an access token to access the data in the API using pre approved Client Credentials.

2.Fetching data from the API

3.Inserting the data to a table in Postgre cluster DB

4.3.2 WORK DONE

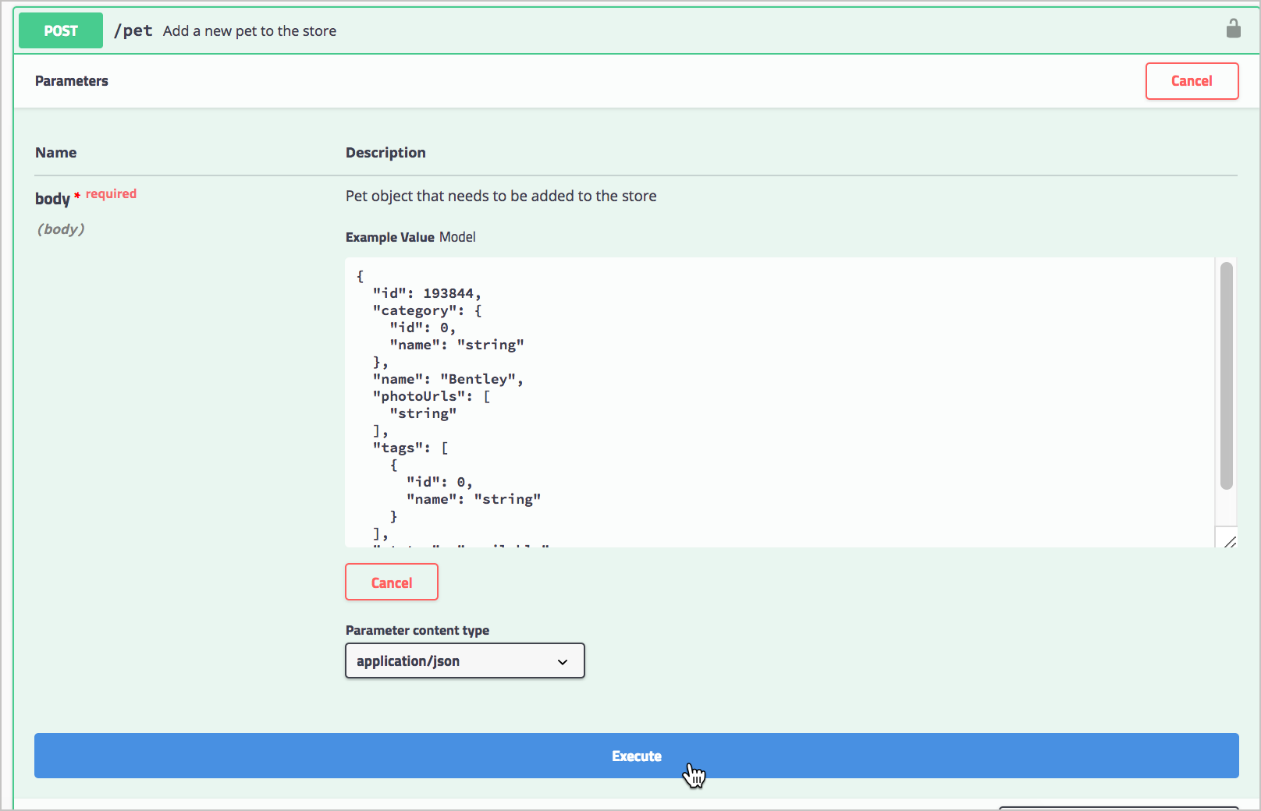
After getting an approval and generating Client Credentials from the respective API managers,I had to look for a way to be able to generate an temporary access token to be provided to the API to get access to an endpoint for me to fetch data from it.This prompted the use of a ClientCredentialsResourceDetails Java object which when used with an OAuth2RestTemplate was able to get us a temporary access token each time we ran the utility using the mentioned scopes,clientId,clientSecret and accessTokenUri.

The next piece of the puzzle was being able to fetch large amounts of data from the API,To do this, I had to use an Http entity which contained headers and some criteria of the columns to retrieve in the POST request body.

The response body is basically a huge JSON Object,consisting of a key called “data” which is basically an array of JSON objects and each JSON Object needs to be added as an individual record into a pre specified table in the Postgres Cluster DB.

This required extensive use of the JSONArray and JSONObject functionalities foung in the Json.Simple library.

Also since this whole process of fetching data and adding it to the DB was to be done everyday at a specific time,making sure that the valuedate for each entry is known was an important feature.We used the Local date() function to get the day’s date on which we are running the util and added it as a separate column in each table we created.



Since parsing this huge JSON Object set us up with a huge overhead in terms of time and due to the lack of a functionality similar to what StreamingReader provided for an excel workbook,the performance metric of speed remains a lot slower for this util as compared to the Excel to Postgres utility.

4.3.3 OPTIMIZATIONS MADE

The optimizations I set in place here were the same that I had set in place for the previous utility:

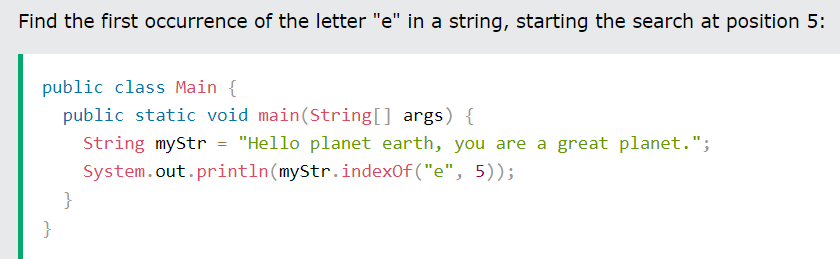
* Following from the principles of a transaction statement,I tried to experiment with passing multiple records using a single insert statement.This seemed to work extremely well and I started experimenting with the number of records that needed to be inserted in a single Insert statement.This testing was important since I had to balance out the read time taken by the utility to parse each row and the time taken by the Insert command to be executed.
* After sufficient enough testing I found the optimal number of records to be 500.With this I could achieve about 1,200 rows inserted per second which was exponentially better than what I was able to achieve before.
* In an attempt to try and do better than this,the next step I took was to implement multithreading.I had read a lot about it in a lot of programming classes but had never experienced its power before.I tried implementing it initially by creating individual runnables but they seemed to be running synchronouslyand therefore not actually using multiple threads at once.The next thing I tried,was creating an Executorservice consisting of a Cached Thread Pool.
* This meant that the insertion time of one batch and the read time of the next batch of rows overlapped resulting in much lesser overhead per batch and me being able to push my performance from a 1,200 rows entered per second to a 2,300 rows being entered per second.
* In an effort to make the tool more configurable we used the @Value annotation of SprinBoot to be able to fetch table name,file name and other such inputs from the User by fetching these values from the application.properties file.
* The utility also has a feature of being able to create a table in the database by reading a configure.xlsx file placed alongside its executable.To create the table,we need to mention the column names and their respective datatypes in the first 2 columns of the excel sheet.Upon running the utility,the code goes throught the excel sheet and adds the configurable table name,column names and their datatypes to a pre configured sql query and then executes the simple create table statement using the JDBC template.
* A copy of this configure file is also saved in a folder called schemas with the name as <table name>\_schema.xlsx for future reference of the user.

4.3.4 ISSUES FACED

The major issue I faced here was that each individual JSON Object that needed to be inserted as a record had a complex structure in itself,i.e had multiple objects inside itself and multiple arrays as well inside it.

Navigating through this required me to get a deeper understanding of the schema of the response body that I was getting from the APIs and this was possible only after digging deeper into the documentations of the respective APIs.

Another issue I was facing here was the insertion of multiple single quotes in a varchar field which I was able to solve using the iterative function String.indexOf() to find positions of multiple occurences of such string literals as shown below:



**Chapter 5**

**My Learnings**

5.1 INDUSTRY EXPERIENCE

Above all technical learnings that I have gathered over the period of the last 4.5 months,what I feel will help me the most as I shift from my academic to my professional career,is the exposure of working in an Industry.

A definition of an Internship that I got off the Internet is :

“Internships can help you gain valuable work experience, fulfill a college requirement and give you material to add to your resume. They can introduce you to many aspects of full-time employment while allowing you to explore your interests and form your personal career goals.”

And I strongly relate to this exact definition of an Internship.This brief internship with Societe Generale helped me in the following ways:

5.1.1 Work Experience:

As a Young Graduate with next to no work experience ,these four and a half months gave me a taste of what it feels like to be an IT professional on a miniature scale.

5.1.2 College Requirement:

I was required to do an Internship of a minimum of 4 months in an organization related to IT organization and Societe Generale was kind enough to offer this one of a kind experience where I could learn,grow ,enjoy and simultaneously fullfill my colleg’s criteria for completion of my B.Tech-IT Programme.

5.1.3 Adding to my Resume:

As a Young Graduate,having just college projects,hackathon participations and some volunteer work was all I could fathom to put into my Resume.This would entail not having a work experience section in my Resume previously,wich now I can proudly fill in as a result of this internship.

5.1.4 Introduction to FTE:

Something that a 4th year engineering graduate is always a bit nervous and anxious about is transitioning from academics to being a working professional.Safe to say that is something that no longer plays on my mind due to this wonderful and enriching time that I have had with Societe Generale.

5.1.5 Mentorship:

The most valuable mentor relationships often start with a personal connection that enhances the experience for both the mentor and the mentee. Working as an intern may allow you to meet a potential mentor naturally and establish a relationship that helps guide your career path.

In-person, you have the chance to reveal your personality and show deference and respect while seeking guidance and input. An internship can create a situation where your relationship develops organically, even if you end up working elsewhere in a full-time position.

5.1.6 Create a Professional Network:

The most successful job searching usually comes from meeting others in your field who can recommend you for open positions. Internships are a practical way to expand your job network. The professionals you will meet might be the most valuable connection to your future jobs, so showing curiosity, enthusiasm and willingness can help professional contacts see your potential.

5.1.7 Secure good references and recommendations:

The supervisors and mentors you meet during your internship can be a valuable reference for you as you pursue a full-time job. The more positive and hardworking you are, the more likely managers are willing to recommend you for open positions. References generated from an internship can be valuable in your job searches because those managers will have known you personally and seen how you contributed to the company. They can speak to specifics and describe your developing skills and how your education prepares you for employment.

5.2 BUILT CONFIDENCE

The transition from college to full-time employment can fill you with excitement, anxiety, hope and ambition all at once. An internship is a good way to fill some transition time with the work you hope to do long-term. It can eliminate some pressure of quickly finding a permanent job and help you apply the skills and knowledge you have been studying into a practical situation.

Internships can show you how a company develops leaders rather than just learning about it in the abstract. You can observe excellent time management skills and make a note of those you want to emulate. You may learn about valuable collaboration and cooperation skills. If you think of an internship as an extension of your education, you are likely to have a whole range of confidence-building attributes to apply to your future employment.

5.3 TECHNICAL ASPECT

5.3.1 Frameworks

As someone who has no work experience it was a very enriching experience to be working with a JAVA Full Stack framework, namely:

* React JS,
* SpringBoot,
* Postgres etc

From the basics of React to using Context to remove disadvantages of Prop Drilling and from the basics of Spring to setting Client Credentials and frequently dealing with Response Entities,the internship was a deeply enriching experience with a steep yet elegant learning curve like no other.

Following are a list of subdomains that I got to work closely with:

1.React JS:-

* Fundamentals of JS
* Props
* Prop Drilling
* Context
* Hooks
* Conditional Rendering
* Refs
* Router
* Bootstrap
* State
* Fragment

2.Spring:-

* Maven
* Containers
* Dependencies
* MVC
* Auto Configuration
* Starter Dependency
* Actuator
* Data JPA
* Properties Files
* YAML Support
* JDBC  
  H2 Database
* CRUD Operations
* RESTful WebServices
* Microservices

3.Postgres:-

* Keys
* Constraints
* Indexes
* Privileges and Passwords
* Literals
* Queries
* JOINS
* Data Types
* JSONB
* Triggers
* Views

**Chapter 6**

**Conclusion & Future Work**

To conclude,I would like to mention my failures, my successes and scope for improvement in the work done.

Starting with the successes, there was a lot to be proud of. Even thought the tools I created ,whether it be backend or frontend , did not require a very high level of technical know how, but for someone with almost negligible technical know how and no prior work experience, being able to deliver UIs and utilities according to a set of business requirements and within the deadlines set and most importantly within the standards set by the company was something for me to be very proud of.

Another aspect that I would consider a huge success for me is the ability to solve problems individually ,and more importantly approaching people way senior than me to offer their advice and implement and learn from it. The aspect of solving problems individually is something that I feel programmers and developers inculcate only with work experience.This ability to do so builds confidence in their skills.

The aspect of approaching colleagues who are senior to me teaches me of how to work as a part of a team,where these situations are a regular instance.This taught me to respect boundaries and more importantly to start valuing others time.

Coming to the technical aspect of my projects ,I was able to create fully functioning SPA standalone screens in react JS for Onboarding of New Joiners. I was also able to make small yet powerful Java utilities to push data from APIs having complex schemas into a table in a Postgres cluster DB with good performan metrics and in accordance with all business requirements. I was also able to create a fully configurable Java tool to read data from any excel,json or csv file and store it into a table in Postgres DB Cluster.

Failures were far and few but one notable one was the inability of my Java utility to extract only desired columns from a excel or a csv file. Meaning either all the columns will be inserted or none will.

Coming to the future scope of improvement , creation of a VM instance on the company servers to automate running of the utilities using a Job Scheduler is something that can be worked upon.

Even though the utility starts working at the click of a button, but to remove human intervention would be the next step in fully automating the workflow. My skills in this field are next to none , so I have left it to the experts to come up with the implementation of this idea.

Also connection of the standalone Onboarding tool UI to a suitable backend is out of my technical skill pool , and I am sure someone with the appropriate know how will work on it and maybe we’ll be seeing my UI used to onboard New Joiners.

**Appendices**

Appendices are provided to give supplementary information, which is not included in the main text may serve as a separate part contributing to main theme.

* + - Appendices should be numbered using Arabic numerals, e.g. Appendix 1, Appendix 2 etc.
    - Appendices, tables and references appearing in appendices should be numbered and referred to at appropriate places just as in the case of chapters.
    - Appendices shall carry the title of the work reported in it and the same title shall be used in the contents page also.

REFERENCES

Eckart, C. (1951). Surface waves on water of variable depth, Wave Rep. 100, Scripps Inst. of Oceanogr., Univ. of California, pp.99. <For referring reports>

Hasselmann, K., W.H. Munk and G.J.F. MacDonald (1963). Bispectra of Ocean Waves, In: M. Rosenblatt (ed.), Time Series Analysis, John Wiley Sons, p. 125, NewYork. <For referring from books edited from a collection of different papers>.

Stoker, J.J. (1957). Water Waves. Interscience, New York. p. 520.

<For referring book>

Tatavarti, Rao V.S.N. and D.A. Huntley (1987). Wave reflection at Beaches. Proc. Canadian Coastal Conference, Quebec City, pp. 241-255, Canada.

<For referring conference proceedings>

Wallace, J.M. and R.E. Dickinson (1972). Empirical orthogonal representation of time series in the frequency domain, Part I: Theoretical considerations, J. App. Meteorology, Vol. 11, No. 6, pp. 887-892. <For referring journals>

<Times New Roman, Font 12>