



Msc.IT Internship Report

Aniket Gajjar

(201812004)

Supervisor

Prof. Saurabh Tiwari
(MSc-It Coordinator)

On-Site Supervisor

Mr. Vikas Raj Karadia
Aws & Java Developer
(Infoobjects Inc. - Jaipur)



Dhirubhai Ambani Institute of Information and
Communication Technology

Table of Content

<i>Sr No.</i>	<i>Description</i>	<i>Page No.</i>
1	Introduction	
2	Abstract	
3	Context Diagram	
4	Scope of the Project	
5	Detailed Training Report	
6	Design Contribution	
7	Tools, Technologies, APIs and Libraries used	
8	Testing Strategies	
9	Innovative Contribution	
10	Lessons Learnt	
11	Bibliography	

Introduction

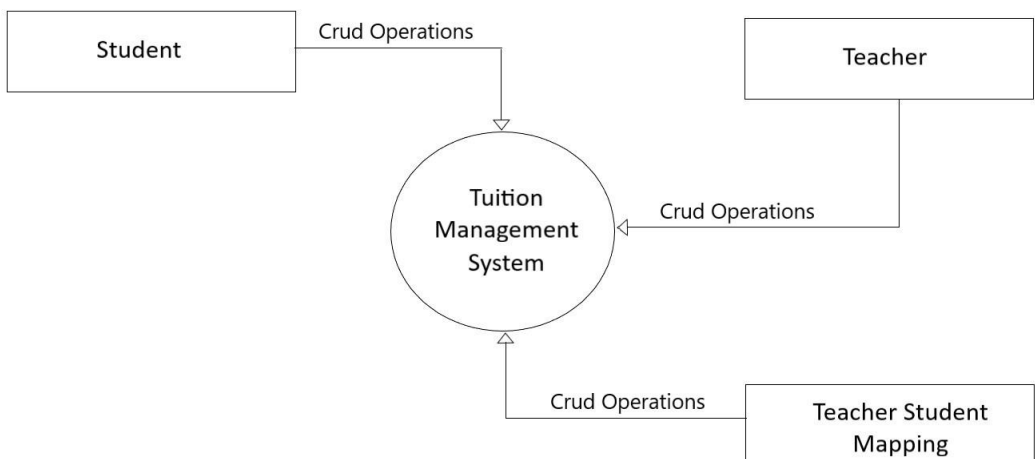
- Infoobjects is an *award-winning* consulting company with focus on Big Data, IoT and public cloud.
- Company's core expertise on big data side lies in Apache Kafka, Apache Spark and Apache Hadoop stack. We are pro public cloud and have tons of expertise in cloud migration.
- Infoobjects is a consulting company that helps enterprises transform how and where they run applications and infrastructure.
- From strategy, to implementation, to ongoing managed services, Infoobjects creates tailored cloud solutions for enterprises at all stages of the cloud journey.
- Company's Engineers are cloud experts who are transforming business through consultative engagements, intellectual property, executive briefings, and thought-leading events.
- Our flexible approach to cloud adoption and acceleration is designed with enterprise needs in mind. We create solutions that address complex challenges such as :
 - Leveraging existing investments while adopting emerging technologies
 - Building consensus on an approach and timeframe for cloud adoption across the organization
 - Addressing organizational changes and skill gaps created by cloud transformation

Website	http://www.infoobjects.com
Industries	Information Technology and Services
Company size	201-500 employees
Headquarters	San Jose, CA
Type	Privately Held
Founded	2005
Specialties	Kafka, Hadoop, AWS, Spark, Azure

Abstract

- This document presents a summary of the internship period undertaken as part of my fourth-semester project.
- This document gives an insight into the project that I worked on as a part of my internship period.
- This document includes the introduction, scope, contribution, relative diagrams and other project details.
- My Internship is going through different stage of training of 6 months.
- I have worked on simple crud dummy Project During my internship period. So I have created this report in context of simple dummy Project.
- I learnt so much in my internship even if it is a Dummy Project that I depicted in this report.

Context Diagram



Scope of the Project

- Tuition Management System is a simple crud Application which are Performed Crud Operations using different Ways.
- Tuition Management System is having three entities : Teacher, Student and Teacher Student.
- Details of Operation that Each Entity performed are as below :
 - Teacher
 - Insert Teacher
 - Delete Teacher
 - Update Teacher
 - Find All Teacher
 - Find Teacher By Teacher Id
 - Student
 - Insert Student
 - Delete Student
 - Update Student
 - Find All Student
 - Find Student By Student Id
 - Teacher Student Mapping
 - Insert Teacher Student Mapping
 - Delete Teacher Student Mapping
 - Find All Teacher Student Mapping
 - Find Teacher Student Mapping By Teacher Id
 - Find Teacher Student Mapping By Student Id
 - Find all Students for Mapping (Students which are not Mapped with all Teachers)
 - Find all Teachers for Mapping for Specific Student Id(Teachers which are not Mapped with Specific Teacher Id)

Detailed Training Report

- My internship are going through different stages . The detailed report is as below :
 1. Core Java and Some Advanced topics of java Brush Up
 - In this stage of internship, I read Core java like static, final, abstract, Interfaces etc. and some advance topics like collections, Generics, Collections etc.
 2. Core Java Dummy Console Application
 - In this stage of internship, I have to create simple Console Application .
 - Applications are performing Insert, Update, Delete, find by Id and find All for Only Student Entity.
 - I Have Implement this project using *MVC design pattern*. I have break all code into different layers like DAO Layer, DTO Layer, Service Layer, View Layer, Utils Layer etc.
 - Student data are stored in temporary storage Hash Map.
 - I Have to create *Generic Dao Layer* using Generics.
 - I Have used Interfaces for Abstraction.
 - The whole project are created using maven and I have to add jars manually and attach to project using pom.xml.
 - I also create one generic toString method using reflection which dynamically print all field of particular class. You just have to pass reference into it. If you add any extra field it will automatically print value of that field dynamically.
 3. Core Java Dummy Console Application (with added new features)
 - In this stage of internship, I have Implemented same application with some new features.
 - The new Entity added called Teacher and I have to perform same operations like student but difference is data stored in database.
 - I have to implement DAO layer for Teacher using JDBC.
 - I also have to map Students with teacher using Teacher Student mapping Table in database.

- Every time Application Closed Mapping data need to be empty because Student data stored into temporary storage.
 - I have Implemented *Singleton Design Pattern* for database Connections.
4. Advanced Java web Project(with some other added features)
- In this stage of internship, I have Implemented same application with some new features.
 - Student Entity which are stored into temporary storage are now going to store into mysql database.
 - I have created *one generic DAO class* which are creating SQL query dynamically according data that comes to class and only one method per operation. I have used hash map and Jackson API to implement this feature.
 - I Have also make my Singleton class of database connection *Thread Safe* With the use of *Volatile* Keyword.
 - I have used *UUID* instead of simple integer id using UUID generator.
 - I Have used Servlet and .html file for controller and UI.
 - I used *HTML 5* and *CSS 3* in UI side.
 - I also created one generic show All page which are used to show all data in UI side . I also used data table to display show all data.
 - I also used snake case in database column and camel case in DTO field name. so you can not directly create SQL query . you have to do some conversion.
5. *Spring MVC* Dummy Project
- In this stage of internship, I have Implemented same application with some new features.
 - I have Converted Project created in earlier stage into *Spring MVC* with *hibernate*.
 - I have used Hibernate for mapping Entity class with database.
 - I also used JSP at UI side instead of html file.
 - I have used *Expression Language* for better performance of loading .
 - I used annotation based Spring Controller for handling HTTP request using different HTTP methods.

- I have create one generic DAO class using hibernate template and generics. This class will perform operations according to Objects comes into methods.
- I also Implement logging Mechanism using log4j API. I have created different appenders for different logging level, create log files for info warn and error message and one for all logging messages. API create new log file dynamically if it exceed its size.
- I also documents java files and method using java API and comments.
- I have also used *AJAX* for dynamically loading data of specific page without loading page.
- I used *jQuery* for form validations.

6. *Spring MVC* Dummy Project with *Angular JS* as a Front End

- In this stage of internship, I have Implemented same application with some new features.
- I have to implement same project with *Angular JS* as a Front end.
- I have to used *Angular JS* Http API to request to the *Spring Controller*.

Design Contribution

- I will share some of Design Screenshots that are Implemented in Different stage of Internship.
- The Following are Designs that I have created into 2nd and 3rd stage of training.

```

TmsStart [Java Application] C:\Program Files\Java\jre-9.0.4\bin\javaw.exe (09-May-2020, 7:54:27 pm)

-----
                        Tuition Management System
-----

1) Student
2) Teacher
3) Exit

Enter Choice :
1
|

1) Insert Student
2) Update Student
3) Delete Student
4) Find Student
5) Find all Student
6) Exit

Enter Choice :

```

```

Console Progress Problems
TmsStart [Java Application] C:\Program Files\Java\jre-9.0.4\bin\javaw.exe (11-May-2020, 9:53:41 am)

1) Insert Student
2) Update Student
3) Delete Student
4) Find Student
5) Find all Student
6) Exit

Enter Choice :
5
Students :

Student :

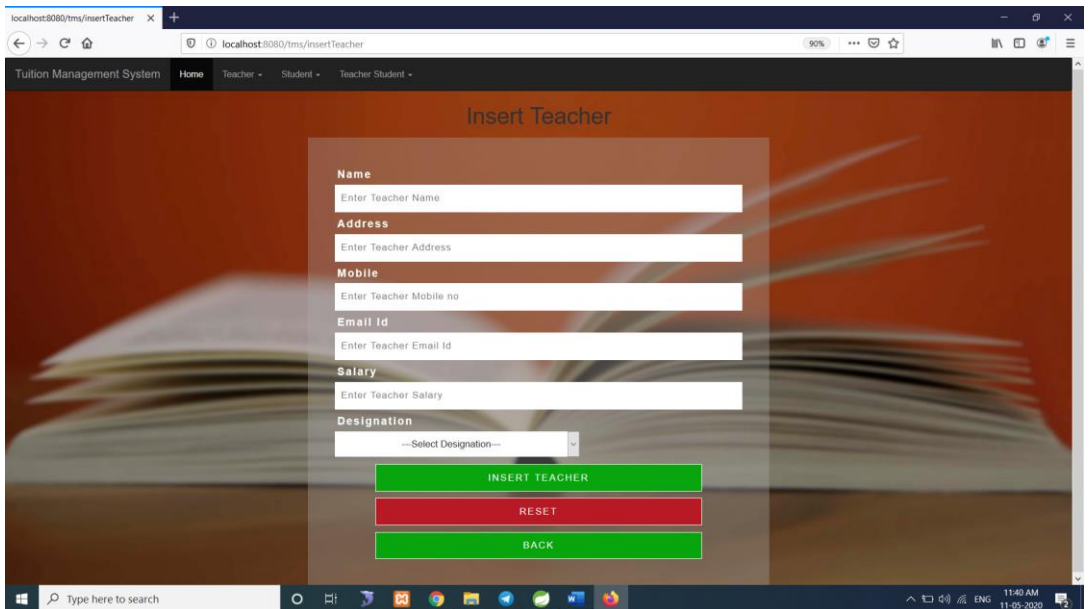
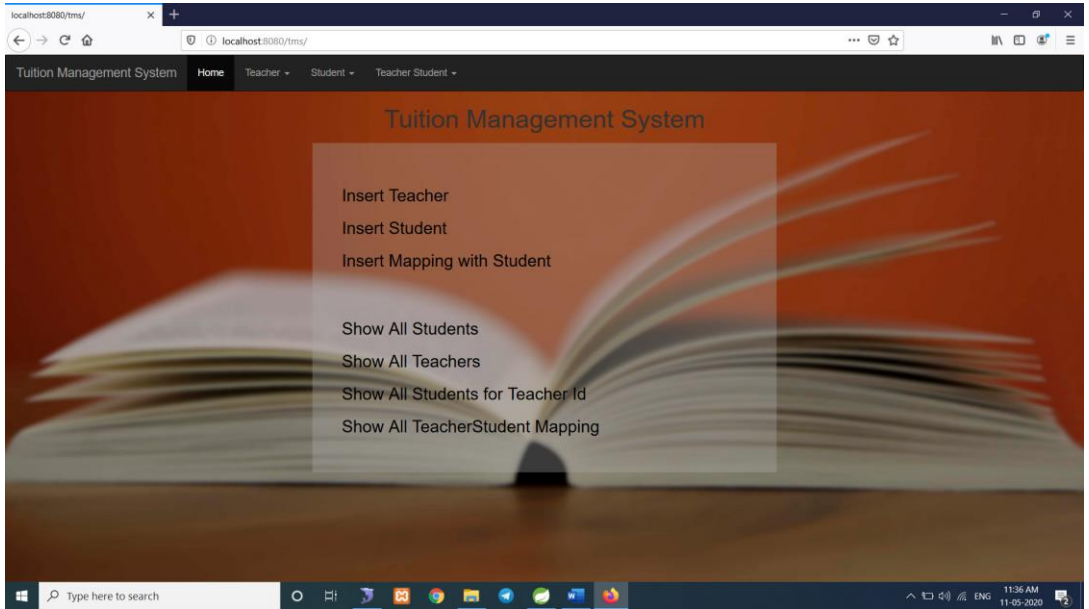
studentId : 2df38327-1f29-4e5a-b383-271f294e5abf
studentName : Harsh
studentAddress : gandhinagar
studentEmailId : kalpana@gmail.com
studentGender : MALE
studentMobile : 9662300408
studentParentName : abd
studentParentMobile : 9106447228
studentParentEmailId : harsh@gmail.com
studentReferenceName : yesha

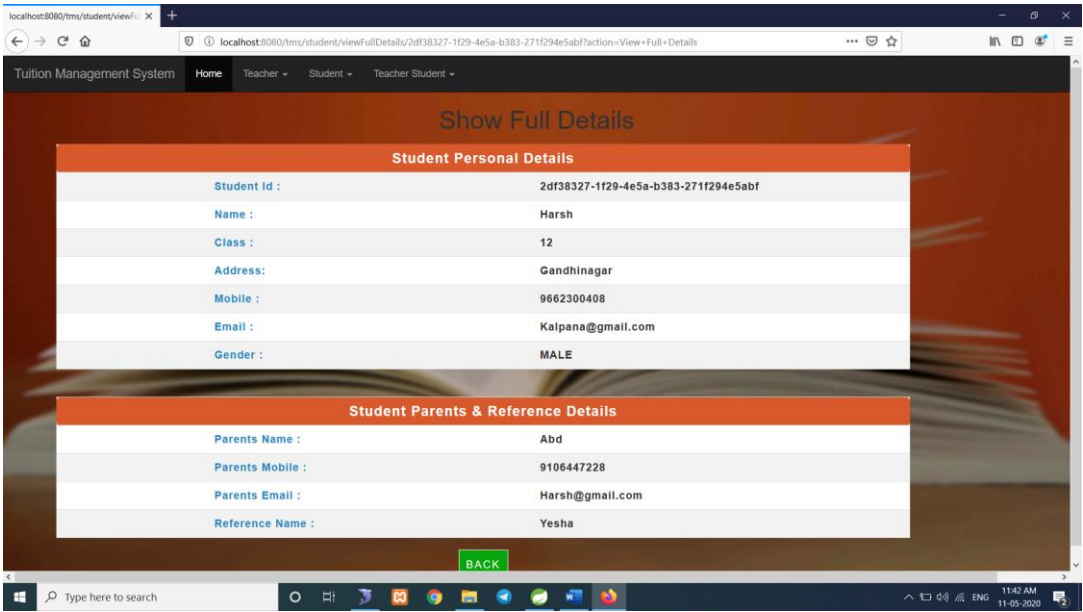
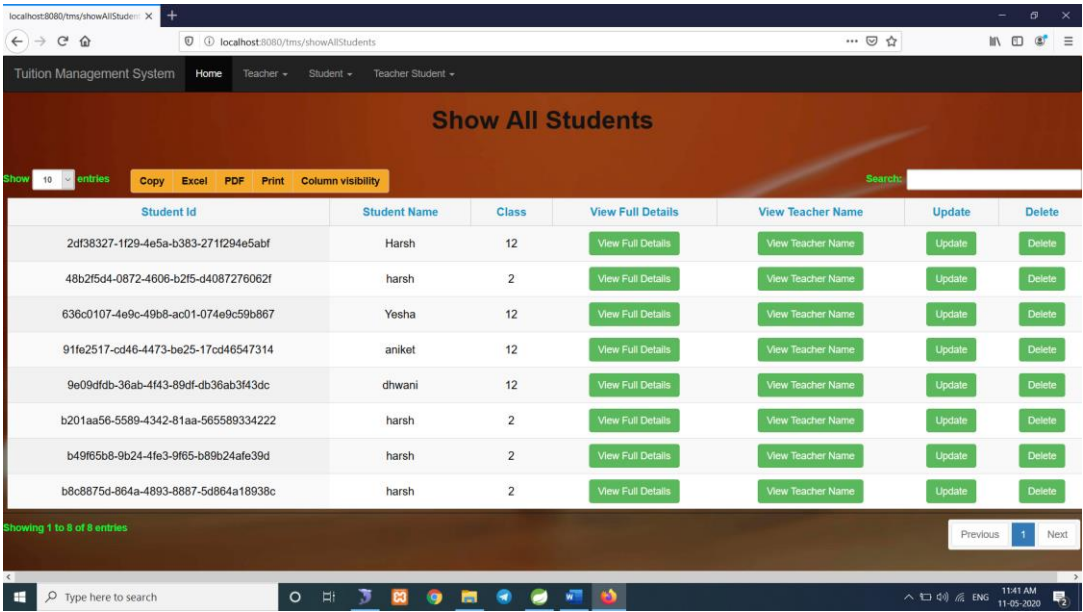
Student :

studentId : 48b2f5d4-0872-4606-b2f5-d4087276062f
studentName : harsh
studentAddress : ahmedabad
studentEmailId : harsh@gmail.com
studentGender : MALE
studentMobile : 9662300408
studentParentName : kalpanaben
studentParentMobile : 9662300408
studentParentEmailId : kalpana@gmail.com
studentReferenceName : yesha

```

- The Following are Designs that I have created into 4th, 5th and 6th stage of training.





Tools, Technologies, APIs and Libraries used

Tools :

- Eclipse
- IntelliJ Idea
- MySQL Server
- MySQL Workstation
- SqlYog
- Git bash
- Postman

Technologies :

- Java
- HTML 5, CSS 3, Java Script, Ajax
- Git
- JSP, JSP Expression Language
- Angular JS

APIs :

- Java Persistent API
- Java Transaction API
- Java Documentation API
- Log4j Logging API
- Angular JS Http API
- Jackson API

Libraries :

- JDK
- Spring Framework
- Hibernate
- Java Persistent
- Log4j
- Commons logging

Testing Strategies

- Testing Strategy that I undertook was unit testing. It was done using test each class code with giving appropriate input and check its output whether it is right or not in core java project. I used postman to test API that I have created in spring project.
- As soon as I develop a API, the HTTP request was simulated in the environment of Postman and the correctness of response was measured based on the desired output.

Innovative Contribution

- I have done three POC(Proof of Concept) work with my internship Project. The details of POC Work are as below :

1. Prisma.io

- Prisma is GraphQL API which replaces traditional ORMs and simplifies database workflows.
- Access: Type-safe database access with the auto-generated Prisma client (in JavaScript, TypeScript, Go)
- Migrate: Declarative data modelling and migrations (optional)
- Manage: Visual data management with Prisma Admin
- It is used to build GraphQL, REST, gRPC APIs and a lot more. Prisma currently supports MySQL, PostgreSQL, MongoDB.
- Prisma is a standalone infrastructure component that sits on top of your database. You're then using a Prisma client (which is available in various languages) in your application server to connect to Prisma.
- This enables you to talk to your database(s) through a simple and modern API ensuring highly performant and secure database access.

- Prisma is the perfect tool for building GraphQL servers. The Prisma client is compatible with the Apollo ecosystem, has default support for GraphQL subscriptions and Relay-style pagination, provides end-to-end type safety and comes with a built-in data loader to solve the N+1 problem.
- Type-safe database access thanks to the custom and auto-generated Prisma client.
- Simple and powerful API for working with relational data and transactions.
- Visual data management with Prisma Admin.
- Prisma unifies access to multiple databases at once (*coming soon*) and therefore drastically reduces complexity in cross-database workflows.
- Realtime streaming & event system for your database ensuring you're getting updates for all important events happening in your database.
- Automatic database migrations (optional) based on a declarative data model expressed using GraphQL's schema definition language (SDL).
- Other database workflows such as data import, export & more.

2. Load Balancer on AWS Lambda

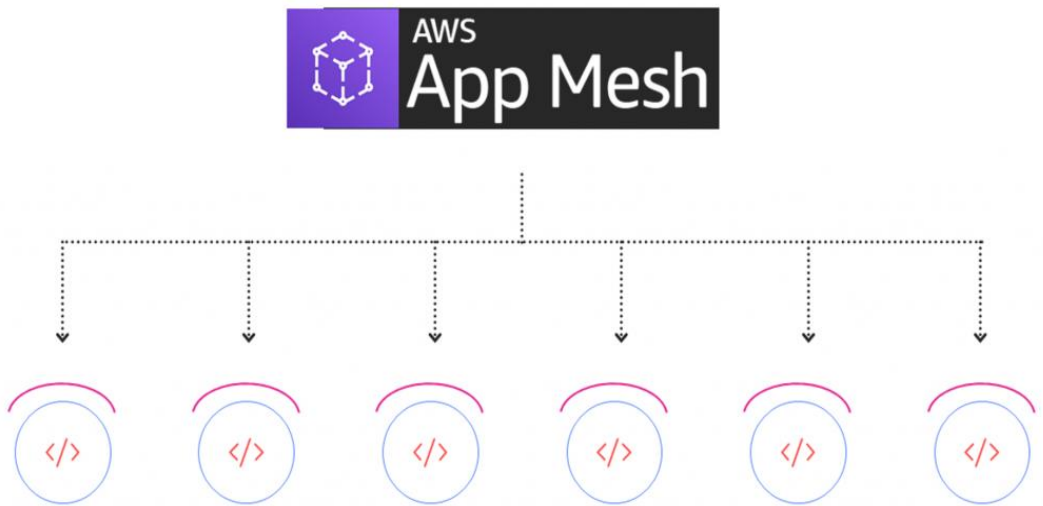
- Application Load Balancers (ALBs) now support AWS Lambda functions as targets. Build websites and web applications as serverless code, using AWS Lambda to manage and run your functions, and then configure an ALB *to provide a simple HTTP/S frontend for requests coming from web browsers and clients*.
- You can register your Lambda functions as targets and configure a listener rule to forward requests to the target group for your Lambda function.
- When the load balancer forwards the request to a target group with a Lambda function as a target, it invokes your Lambda function and passes the content of the request to the Lambda function, in JSON format.
- Application Load Balancer now provides you a single HTTP endpoint for all requests whether they are served from EC2, containers, on-prem servers or Lambda functions.

- With the Application Load Balancers' support for content-based routing rules, you can also route requests to different Lambda functions based on the request content.
- The Lambda function and target group must be in the same account and in the same Region.
- The maximum size of the request body that you can send to a Lambda function is 1 MB. For related size limits, see HTTP Header Limits.
- The maximum size of the response JSON that the Lambda function can send is 1 MB.
- WebSocket are not supported. Upgrade requests are rejected with an HTTP 400 code.

3. AWS App Mesh

- AWS announced AWS App Mesh, a service mesh that provides application-level networking. App Mesh makes it easy for your services to communicate with each other across multiple types of compute infrastructure, including Amazon EKS, Amazon ECS, Kubernetes and Amazon EC2.
- App Mesh standardizes how your services communicate, giving you end-to-end visibility and ensuring high availability for your applications. Service meshes like App Mesh help you run and monitor HTTP and TCP services at scale.
- Using the open source Envoy proxy, App Mesh gives you access to a wide range of tools from AWS partners and the open source community. Because all traffic in and out of each service goes through the Envoy proxy, all traffic can be routed, shaped, measured, and logged.
- This extra level of indirection lets you build your services in any language desired without having to use a common set of communication libraries.
- With a service mesh, you can decouple your microservices' observability, analytics, and routing logic from the underlying infrastructure and application layers.
- App Mesh implements this sidecar proxy via the production-proven Envoy proxy. Envoy is arguably the most popular open-source service proxy.

- In the following diagram, a sidecar runs alongside each container in your application to provide its proxying logic, syncing each of their unique configurations from the App Mesh control plane.



- Each one of these proxies must have its own unique configuration ruleset pushed to it to operate correctly.
- To achieve this, DevOps teams can push their intended ruleset configuration to the App Mesh API.
- From there, the App Mesh control plane reliably keeps all proxy instances up-to-date with their desired configurations.
- App Mesh dynamically scales to hundreds of thousands of pods, tasks, EC2 instances, and Lambda functions, adjusting configuration changes accordingly as instances scale up, down, and restart.

Lessons Learnt

- learn a lot of new things in these four months of my internship. Also the things which I had learnt in classroom and books were put to test with a lot of practical implementation.
- *Code Design* and *Design pattern* are crucial part of good product.
- learn how important documentation is before starting with coding.
- learn how to use Java documentation API.
- learn how to apply knowledge that you learn while studying.
- How to work effectively in Professional environment.
- How to write code Structurally Correct.
- learn new Java frameworks like hibernate , Spring.
- Learn how to understand, debug and work on the existing code which consists of various integrated technologies and libraries.
- Learn JPA.
- Learn MVC Framework and different libraries.
- Quick Learning and finding the resources or the means to meet requirements on time.
- Learn UI Technologies like HTML 5, CSS 3, JavaScript, jQuery and Angular JS.
- Learn use variables String instead of hard code it.
- Learn how to reuse code using method.
- Use String formator instead of using concating strings.
- Learn how to use logging API .
- Learn how to use Transaction In java.
- Learn how to done validation using jQuery.
- Learn various technologies related to AWS clouds.
- Learn how to use GraphQL to improve performance of Application.

Bibliography

- I have used following resources during my Internship for work.

- 1) Tutorialspoint
- 2) w3schools
- 3) geeksforgeeks
- 4) mkyong
- 5) spring.io
- 6) Aws Documentation
- 7) Quora
- 8) Github
- 9) Google Translate
- 10) <https://git-scm.com/docs/git-branch>
- 11) <https://www.programiz.com/>
- 12) Telusko Channel (<https://www.youtube.com/user/javaboynavin>)
- 13) Java Brains Channel (<https://www.youtube.com/user/koushks>)
- 14) Edureka Channel (<https://www.youtube.com/user/edurekaIN>)
- 15) gontuseries Channel (<https://www.youtube.com/user/gontuseries>)

- These are some of the resources that I have used during my internship .