Connect iPass & Hire 2 Retire

PREPARED BY

Harsh Udai

(2018 B. Tech CSE 106)

FACULTY GUIDE

Dr. Gireesh Kumar Kaushik



Department of Computer Science Engineering
Institute of Engineering and Technology (IET)
JK Lakshmipat University Jaipur

July 2022

Connect iPass & Hire 2 Retire

Practice School-2

Submitted in partial fulfilment of the requirements

for the degree of

Bachelor of Technology in Computer Science Engineering

By:

Harsh Udai

(2018 B. Tech CSE 106)

Faculty Guide:

Dr. Gireesh Kumar Kaushik



Department of Computer Science Engineering
Institute of Engineering and Technology (IET)
JK Lakshmipat University Jaipur

July 2022

CERTIFICATE

This is to certify that the Practice School-2 project work entitled "**Connect iPass & Hire2Retire**" submitted by **Harsh Udai (2018 B. Tech CSE 106)**, towards the partial fulfilment of the requirements for the degree of **Bachelor of Technology in Computer Science Engineering** of JK Lakshmipat University Jaipur is the record of work carried out by them under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for Practice School-II examination.

Prof. Dr. – Sanjay Goel
Director
Institute of Engineering & Technology
JK Lakshmipat University Jaipur

Dr Sonal Jain
Department of Computer Science Engineering
Institute of Engineering & Technology (IET)
JK Lakshmipat University Jaipur

Dr Gireesh Kumar
Department of Computer Science Engineering
Institute of Engineering & Technology (IET)
JK Lakshmipat University Jaipur

Date of Submission

Acknowledgement

I would like to express our heartfelt gratitude to Dr. Gireesh Kumar Kaushik, Professor, Department of Computer Science & Engineering for his valuable time and guidance that made the project work a success. I would also like to thank Mr. Shashank Garg, for their valuable input to the project work (as an Industry Mentor).

I would also like to thank sincerely to Prof. Dr Sonal Jain, Head of Computer Science Department and Prof Sanjay Goel, Director, Institute of Engineering and Technology, JK Lakshmipat University for giving us a platform to carry out the project.

Sincerely yours, Harsh Udai

Harsh Udai, 2018 B. Tech CSE 106

List of Figures:

Figure 1. Organization Logo 9
Figure 2. Connect iPass 11
Figure 3. Rabbit MQ 12
Figure 4. Java Spring Framwork 13
Figure 5. Microservices Architecture 14
Figure 6. Node js 15
Figure 7. SQL 16
Figure 8. Git and GitHub 17
Figure 9. Docker 18
Figure 10. Kubernetes 19
Figure 11. Jenkins 20
Figure 12. Hire 2 Retire 21
Figure 13. H2R Automation 23
Figure 14. Development Stages. 24
Figure 15. Jenkins and Kubernetes Workflow 25
Figure 16. Node js 29
Figure 17. Event Loop Node js 30
Figure 18. Microsoft Graph API 34
Figure 19. Rest Architecture 36
Figure 20. Git Branches 37
Figure 21. Git Merge Conflicts 38
Figure 22. Dockerization 40
Figure 23. Image Build with Jenkins 42
Figure 24. Jenkins and Docker 44

Contents

RoboMQ Pvt. Ltd. 9
Project Definition and Plan 10
Tools and Technology 11
Connect iPass 11
Messaging Broker: 12
RabbitMQ 12
Backend Development 13
Java 13
Java Spring 13
Microservices 14
Principles of Microservices 14
Nodejs 15
Rest API – (Rest Architecture) 15
Databases: 16
MYSQL 16
CI – CD & DevOps 17
Git and GitHub 17
Docker 18
Kubernetes 19
Jenkins 20
Hire 2 Retire 21
Benefits of using Hire to Retire Product 22
Backend Development 24
Workflow 25
Implementation of Learnings 26

Bug Solving 27
Scripting 28
Node js 29
Threading and Scheduling 31
New State 31
Runnable State 32
Running State 32
Blocked State 32
Dead State 32
Feature Development 33
Microsoft Graph API 34
Versioning of the project 37
Commit 37
Pull 37
Push 37
Handling Merge conflicts 38
One solution to handle such cases: 38
Dockerising the Application 39
Advantages of using docker: 39
How we dockerise the Application 40
Building docker image with Jenkins 41
Website Handling with WordPress 45
WordPress 45
Frontend: 46
Cascading Style Sheet: 46
Backend: 47
PHP: 47

Testing of the features and finalizing the test cases ____ 48
Research on Topics for new implementation ____ 48
Summary ____ 49
References ____ 50

RoboMQ Pvt. Ltd.



Figure 1. Organization Logo

RoboMQ is an API and data integration company behind some of the leading-edge technologies and platforms that enable enterprises to build business processes and automate their operations using best of breed cloud, SaaS, and on-premises applications that they already own.

RoboMQ accelerates digital transformation to create a competitive advantage using the best-of-the-breed SaaS, cloud, on-premises apps, and IoT. Its Pre-packaged integration solutions (PIPs), Connect iPaaS, and Hybrid Integration Platform (HIP) provides intuitive, UX-based API and data integration for the line of business and technology users.

Connect iPaaS provides a modern, no-code, intuitive user experience (UX) for API and data integration with SaaS, on-premises and cloud services including, but not limited to, Microsoft Dynamics 365, ServiceNow, Workday, ADP, Salesforce and other leading CRM, Finance, Sales and Enterprise Resource Planning (ERP) applications.

Our Hybrid Integration Platform (HIP) provides advanced integration capabilities using Kubernetes based service mesh, service bus, and API management capabilities for highly distributed and scalable integration of Cloud, SaaS and on-premises applications in a hybrid and federated cloud setup. Integration workflows can be built leveraging prebuilt smart containers and capabilities & frameworks provided by the platform.

Project Definition and Plan

The profile for which I am working is "Software Development Engineering".

The main area of my work will be in Full Stack Development. Currently the company has organized various sessions and training sessions.

The company uses JIRA for management and respective tickets are created and assigned for all the technology learning.

As the part of development team, we will be creating backend microservices and ecosystem related services.

The main project of our team is "Connect iPass and Hire 2 Retire" where a user can create integration between various ecosystems for example. Microsoft teams, Microsoft Outlook, Dynamis 365, mysql, mongo dB, etc. In Hire 2 Retire there are various HR system are provided and from them we process our data to active directory.

Tools and Technology

Connect iPass

Connect iPaaS provides a modern, no-code, intuitive user experience (UX) for API and data integration with SaaS, on-premises and cloud services including, but not limited to, Microsoft Dynamics 365, ServiceNow, Workday, ADP, Salesforce and other leading CRM, Finance, Sales and Enterprise Resource Planning (ERP) applications.

Connect iPaaS is targeted at line of business users or citizen integrators seeking to build workflows and API integrations in a matter of minutes using a wide array of available API connectors.

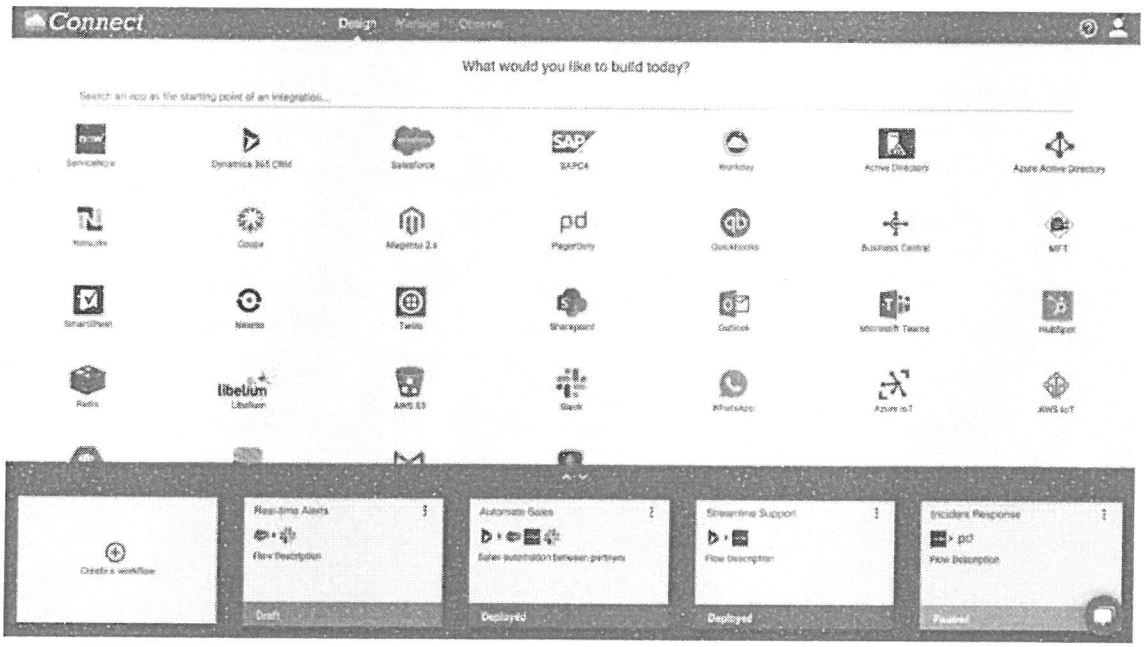


Figure 2. Connect iPass

Through this Connect iPass platform we can create integration flows between various ecosystem like Microsoft outlook and Microsoft Teams and many more ecosystem listed above.

E.g., Whenever we receive an email and notification/message will popup in the Teams channel to inform you and we can also set according to the priority of the message/mail.

Messaging Broker:

RabbitMQ

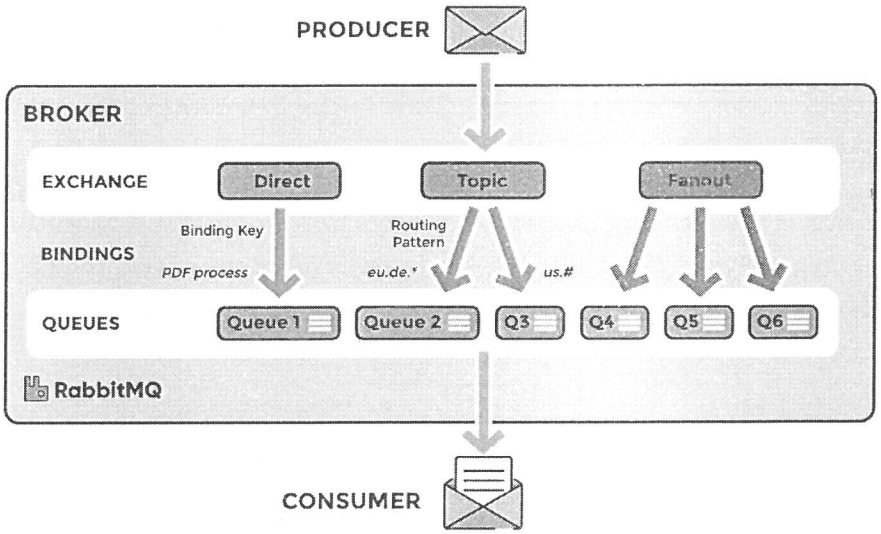


Figure 3. Rabbit MQ

RabbitMQ is an open-source message-broker software (sometimes called message-oriented middleware) that originally implemented the Advanced Message Queuing Protocol (AMQP) and has since been extended with a plug-in architecture to support Streaming Text Oriented Messaging Protocol (STOMP), MQ Telemetry Transport (MQTT), and other protocols.

It is written in Erlang, the RabbitMQ server is built on the Open Telecom Platform framework for clustering and failover. Client libraries to interface with the broker are available for all major programming languages. The source code is released under the Mozilla Public License.

Backend Development

Java

Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is a general-purpose programming language intended to let programmers write once, run anywhere (WORA), meaning that compiled Java code can run on all platforms that support Java without the need to recompile. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. Java was one of the most popular programming languages in use according to GitHub, particularly for client–server web applications, with a reported 9 million developers.

Java Spring



Figure 4. Java Spring Framework

The Spring Framework is an application framework and inversion of control container for the Java platform. The framework's core features can be used by any Java application, but there are extensions for building web applications on top of the Java EE (Enterprise Edition) platform. Although the framework does not impose any specific programming model, it has become popular in the Java community as an addition to the Enterprise JavaBeans (EJB) model. The Spring Framework is open source.

Microservices

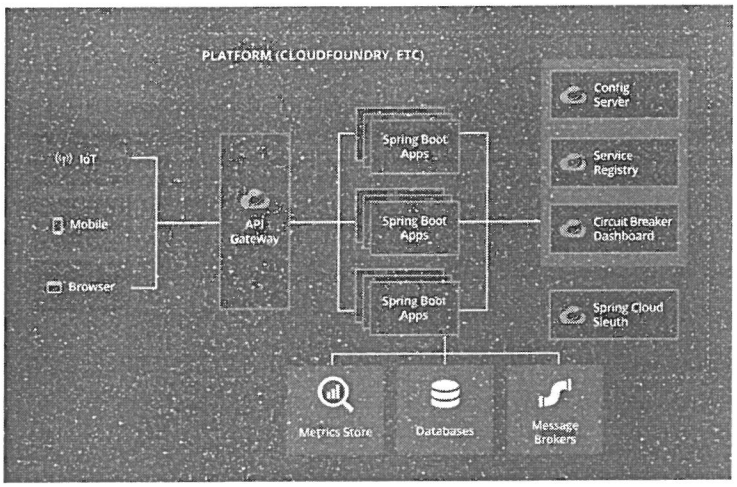


Figure 5. Microservices Architecture

The microservice architectural style is an approach to develop a single application as a suite of small services. Each microservice runs its process and communicates with lightweight mechanisms. These services are built around business capabilities and independently developed by fully automated deployment machinery.

Principles of Microservices

There are the following principles of Microservices:

- Single Responsibility principle
- Modelled around business domain
- Isolate Failure
- Infrastructure automation
- Deploy independently

Nodejs

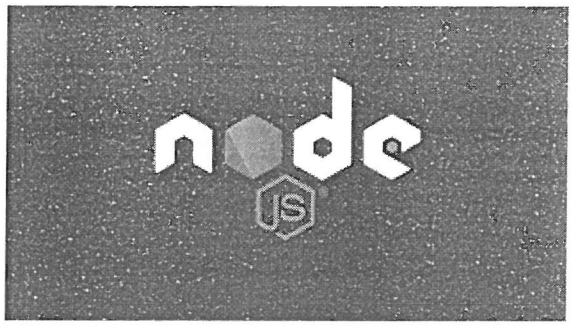


Figure 6. Node js

Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser. Node.js lets developers use JavaScript to write command line tools and for server-side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm, unifying web-application development around a single programming language, rather than different languages for server-side and client-side scripts.

Rest API – (Rest Architecture)

Representational state transfer (REST) is a software architectural style that was created to guide the design and development of the architecture for the World Wide Web. REST defines a set of constraints for how the architecture of an Internet-scale distributed hypermedia system, such as the Web, should behave. The REST architectural style emphasises the scalability of interactions between components, uniform interfaces, independent deployment of components, and the creation of a layered architecture to facilitate caching components to reduce user-perceived latency, enforce security, and encapsulate legacy systems

Methods:

1) GET.
2) POST.
3) PATCH.
4) DELETE.

Databases:

MYSQL

SQL Structured Query Language) is a domain-specific language used in programming and designed for managing data held in a relational database management system (RDBMS), or for stream processing in a relational data stream management system (RDSMS). It is particularly useful in handling structured data, i.e., data incorporating relations among entities and variables. SQL offers two main advantages over older read–write APIs such as ISAM or VSAM. Firstly, it introduced the concept of accessing many records with one single command. Secondly, it eliminates the need to specify *how* to reach a record, e.g., with or without an index.

Originally based upon relational algebra and tuple relational calculus, SQL consists of many types of statements, which may be informally classed as sublanguages, commonly: a data query language (DQL), a data definition language (DDL), a data control language (DCL), and a data manipulation language (DML). The scope of SQL includes data query, data manipulation (insert, update and delete), data definition (schema creation and modification), and data access control.

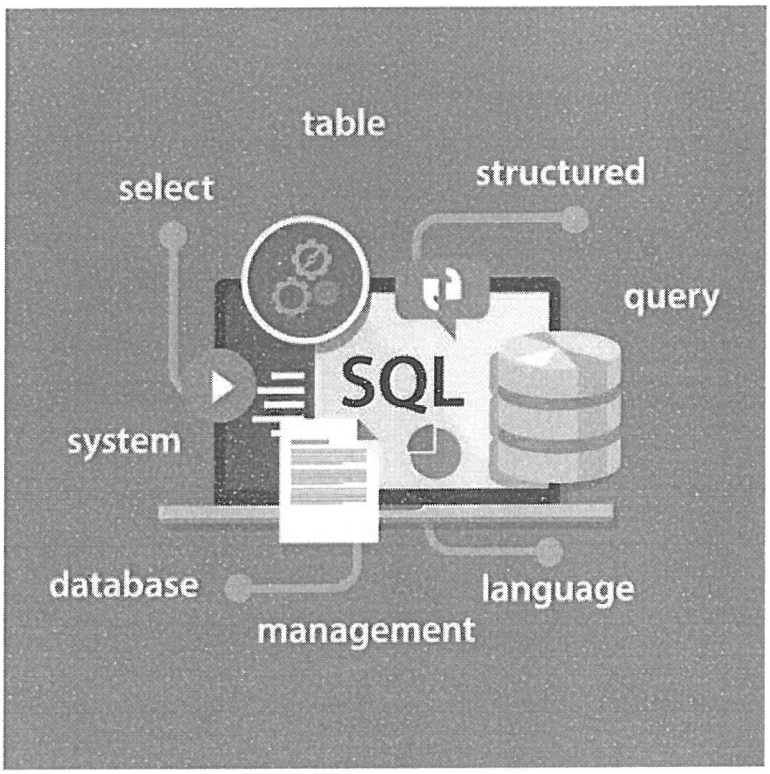


Figure 7. SQL

CI – CD & DevOps

Git and GitHub



Figure 8. Git and GitHub

Git is software for tracking changes in any set of files, usually used for coordinating work among programmers collaboratively developing source code during software development. Its goals include speed, data integrity, and support for distributed, non-linear workflows (thousands of parallel branches running on different systems). Git was originally authored by Linus Torvalds in 2005 for development of the Linux kernel, with other kernel developers contributing to its initial development. GitHub, Inc. is a provider of Internet hosting for software development and version control using Git. It offers the distributed version control and source code management (SCM) functionality of Git, plus its own features. It provides access control and several collaboration features such as bug tracking, feature requests, task management, continuous integration and wikis for every project.

Docker

Docker is a set of platform as a service (PaaS) products that use OS-level virtualization to deliver software in packages called containers. The service has both free and premium tiers. The software that hosts the containers is called Docker Engine. It was first started in 2013 and is developed by Docker, Inc.

Docker can package an application and its dependencies in a virtual container that can run on any Linux, Windows, or macOS computer. This enables the application to run in a variety of locations, such as on-premises, in public or private cloud. When running on Linux, Docker uses the resource isolation features of the Linux kernel (such as cgroups and kernel namespaces) and a union-capable file system (such as OverlayFS) to allow containers to run within a single Linux instance, avoiding the overhead of starting and maintaining virtual machines. Docker on macOS uses a Linux virtual machine to run the containers.

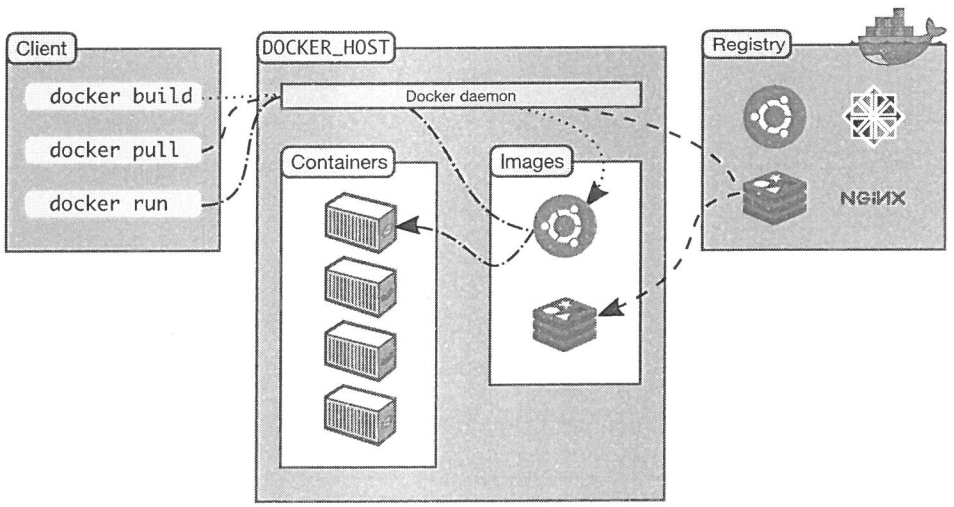


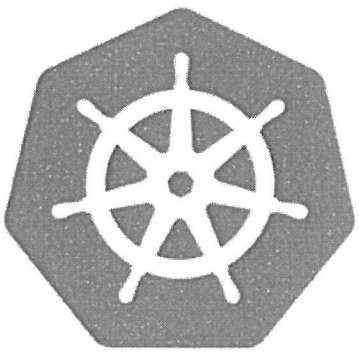
Figure 9. Docker

Kubernetes

Kubernetes is an open-source container orchestration system for automating software deployment, scaling, and management. Google originally designed Kubernetes, but the Cloud Native Computing Foundation now maintains the project.

Kubernetes works with Docker, Containerd, and CRI-O. Originally, it interfaced exclusively with the Docker runtime through a "Dockershim"; however, since 2016, Kubernetes has deprecated the shim in favor of directly interfacing with the container through Containerd, or replacing Docker with a runtime that is compliant with the Container Runtime Interface (CRI).

Kubernetes defines a set of building blocks ("primitives") that collectively provide mechanisms that deploy, maintain, and scale applications based on CPU, memory or custom metrics. Kubernetes is loosely coupled and extensible to meet different workloads. The internal components as well as extensions and containers that run on Kubernetes rely on the Kubernetes API. The platform exerts its control over compute and storage resources by defining resources as Objects, which can then be managed as such.



kubernetes

Figure 10. Kubernetes

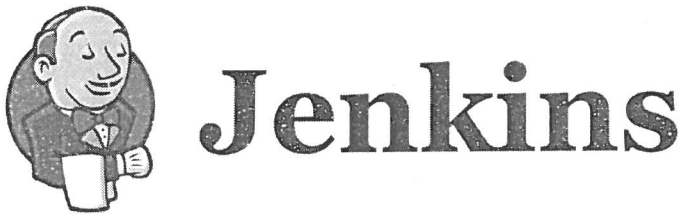


Figure 11. Jenkins

Jenkins is an open-source automation server. It helps automate the parts of software development related to building, testing and deploying, facilitating continuous integration and continuous delivery. It is a sever-based system that runs in servlet containers such as Apache Tomcat. It supports version control tools, including AccuRev, CVS, Subversion, Git, Mercurial, Perforce, ClearCase and RTC and can execute Apache Ant, Apache Maven, and sbt based projects as well as arbitrary shell script and Windows batch commands.

Hire 2 Retire

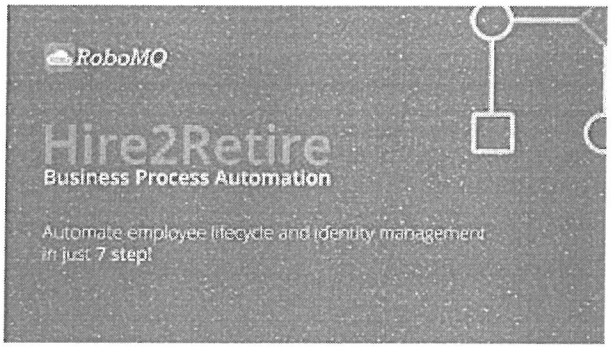


Figure 12. Hire 2 Retire

Hire to Retire is a pre-packaged integration solution for managing employee lifecycle. Business processes that were once performed manually that can now be automated have become essential. The onboarding and offboarding employee process can be achieved automatically. And HR and IT Teams will free up and have more time to focus on other core areas of their business.

After receiving Employee data from HR System, the Hire to retire product will process each record with a series of predefined rules:

1. Map HR profile to AD user identity
2. Define HR user lifecycle events
3. Define rules for OU assignment based on the profile
4. Define rules for Security Group (role-based access)
5. Define rules for Distribution List Memberships
6. Create or update user identity in AD

Benefits of using Hire to Retire Product

1. Enhance employee experience with a faster onboarding process
2. Save operational costs by automating manual system admin work
3. Avoid costly terminations by quickly removing employee access during offboarding
4. Enable single sign on to manage employee access effectively

Hire2Retire design page allows you to define rules for integrating employee data between the HR system and AD. Whenever a new employee report is received, the Hire2Retire workflow will apply the rules to each employee record and create or update profile in your Active Directory. Hire2Retire design page contains steps which help you to manage Employee Lifecycle smoothly.

Following steps have to be configured if you use On Prem Active Directory or Hybrid as your Identity Management System.

1. HR Employee Data Definition
2. Employment Lifecycle Sync
3. Employment Status
4. HR to AD Profile Map
5. Organizational Unit Assignment
6. Security Group Assignment
7. Distribution List Assignment

Following Steps must be configured if you use Azure Active Directory as your Identity Management System.

1. HR Employee Data Definition
2. Employment Lifecycle Sync
3. Employment Status
4. HR to Azure AD Profile Map
5. Microsoft 365 Group Assignment
6. Security Group Assignment

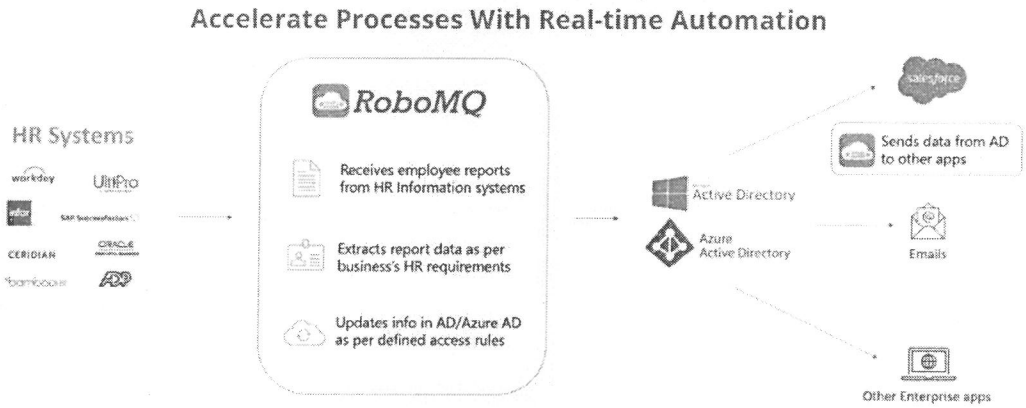


Figure 13. H2R Automation

Backend Development

As part of the backend team, I mostly worked with the Nodejs and Java Spring boot. In Nodejs, basically I used to write the logics to handle the data from the configuration which we got from the frontend and then interpreting the configuration so that it become more readable and easier to understand to our microservices.

As we are using microservices architecture the complete microservices related work is accomplished in the Java. So, the complete architecture of microservice is written in Java spring. There are various benefits of using Java Spring boot, the features which we are using are IOC inversion of control, bean handling, dependency Injection.

So basically, our product HIRE 2 RETIRE serves as a HRIS system, so that we can process all the information of users to active directory without any manual work, we just need a excel sheet and then the product hire 2 retire will do all the task to process information from the excel sheet to the active directory.

There are various ecosystems which serves as a HR system like ADP, Bamboo HR, UltiPro, oracle HCM, Ceridian, Infor HCM, etc. we start by taking permission from the original HR system then started taking data from the system and then process all the information to the active directory.

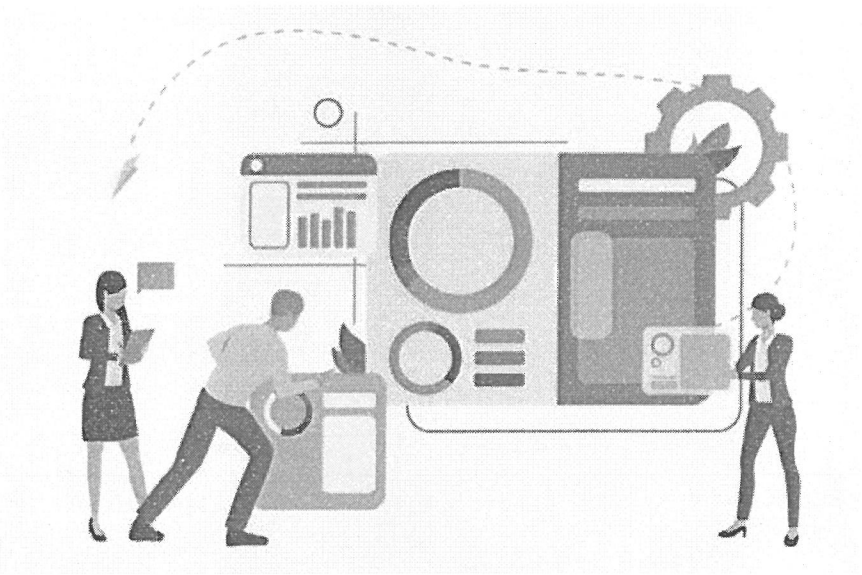


Figure 14. Development Stages.

Workflow

Step 1: We start by making changes to the code let's say we are debugging the code and fixing the bugs, so in this case we just change the buggy code, and this change will be reviewed and then we start with version control system (GIT). First thing which we do is the branch on which we are working and then the important thing is the commit which we made so in commit we just specify the changes which we made in a short message and then we push the changes to the origin so that it became visible to all other fellow developers.

Step 2: After pushing the changes and the complete review done by other developers. We just build the image of the respective microservice so that the changes will be visible to the complete architecture. For this we use Jenkins to build the image of microservice and then we use docker to handle the configuration of the microservices and then it became applied to the latest version of the web interface.

Step 3: There some stages before the final release of the changes, here devops come into picture. So basically, there are some testing environments come into picture. Like first all the changes will be tested by testing team after getting the green flag from the testing team then the final release will be scheduled.

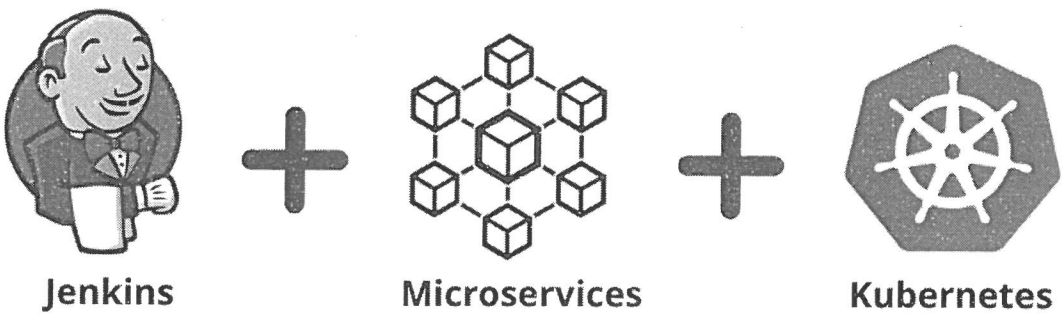


Figure 15. Jenkins and Kubernetes Workflow

Implementation of Learnings

The first phase of learning includes the main and important concepts of Java specifically like encapsulation, inheritance, polymorphism, object-oriented programming and implementing design level question through these concepts and after these spring and its services comes like how to handle beans, how to use xml configuration in order to access another package without explicitly creating the objects and at last important thing is use of data structures. So, after learning and using these topics whenever we write the code for java we always try to things of an optimal and effective solution so that the result we be faster than the usual time. Like implementing a class or extending a class or using abstract class or using an interface so that the security will be achieved as well as the functionality also get setup easily.

After this we start with the most interesting and most demanding language which is JavaScript. In JavaScript we learned about all the topics like hoisting, closures, this keyword, asynchronous behavior, threading, event queue, resource queue, and scoping, etc. So basically, we use JavaScript mostly while writing the NodeJS code and that we uses the integration of various service, like connecting microservices, Redis database, and SQL server, authentication of the webapp and middleware's. So, like whenever we write JavaScript, we always try to thing of the concepts behind the code so that it became more secure and less loosely. For example, like handling the data from the frontend with fetch API or handling the data in more secure way.

Now the important thing is the making ourselves to think in more diverse way of you can say thinking of all the testcases or thinks more ideas for implementing so that if any approach has some flows than we can think of some other approach or some other direction to work on. This phase is more interesting because here we try to think / research over all the possible scenarios and this develops the sense of using technology for solving a problem with different approaches.

Bug Solving

As the part of Backend Team, we also get the bugs related to the product. We start solving the bug by debugging the code for related bug then we start we the buggy part which needed to resolve, for instance I was solving a bug related to Microsoft graph API which is used to create flow between Microsoft teams and Microsoft outlook in connect iPass product. So, for solving this particular bug I learned how Microsoft API works, like how subscription service works and how we access data of an account through rest API's. while solving these bugs also come to know about how we create an endpoint to acknowledge the call of any service. So basically, we need an endpoint to acknowledge the request of any API call of Microsoft Graph API and the bug was also related to that notification URL. In this way I approached the problem/bug and apply learning as well as solution to fix the issues. This practice of solving the bug also helps to make a habit of reading more official documentation and solving the bugs from the same doc.

Same way there are various bugs related to both products (Connect iPass and Hire 2 Retire) were solved to make product more effective and less buggy. One of the major was of Excel function, it is basically a excel function part of the product where we can apply the transformation to the data in the same way we apply to the excel which doing some data manipulation. So there is bug where the condition of excel function fails, so the main reason there was that the type conversion was not correct in node js so that the microservice get wrong input and handle the false information. So, for solving this the need was to change the configuration of the inputs which take from the frontend. It starts from changing the type and make default arguments of the function so that the code still works for the other task and then the issue is specially solved for some cases and the important thing which I have learned here is that we need to specifically define the role of a particular variable and function. The learning was very high while solving the bugs, because at initial phase of any codebase the main thing is to get comfortable with the code workflow and handling small and medium level bugs help to gain more information and understanding the code.

Scripting

Scripting or writing code for one time and for some specific purpose. So basically, as a backend developer there are various reason to write scripts to serve for a special purpose. One was that writing script for the backend compatibility, the main focus for this task is to make some changes to the data part of the database, here all the learning of databases management system also plays an important role, writing complex queries which includes joins, etc. so basically the main change was the format of data in database (SQL) so we have written the script in node js, in order to get the data, update it to the changes required and updating the changes in the databases. Here we did not deal with any service this was the one-time script to directly change into the database.

There are also some product related scripts were written so that, the user interaction will not get disturbed, and user get the expected result. The work which we have done was to add the name of the HR system used with the unique id associated with the user. So, in this we also use direct SQL query to get the data from different tables and then we manipulated the way the data is printed to user UI. So, we just combine the different tables of the user's information which are important for creating the result (name). so, at this stage we also handle configuration with the amazon web services, so after execution of we can say after every change in database we just change with AWS services to ensure that all things work perfectly. So basically, this was also named as backward compatibility script the main focus which we have worked upon is basically with the connection and connection data of the user, we just added more information to the connection data of the user so that at the end the services get the desired results directly.

Writing direct scripts or doing the direct changes to the products requires more attention because we are making direct changes to databases, or it may affect the live performance of the product. So the learning while writing the script was that, writing some peace of code for a particular work is way more tuf than writing general code.

Node js

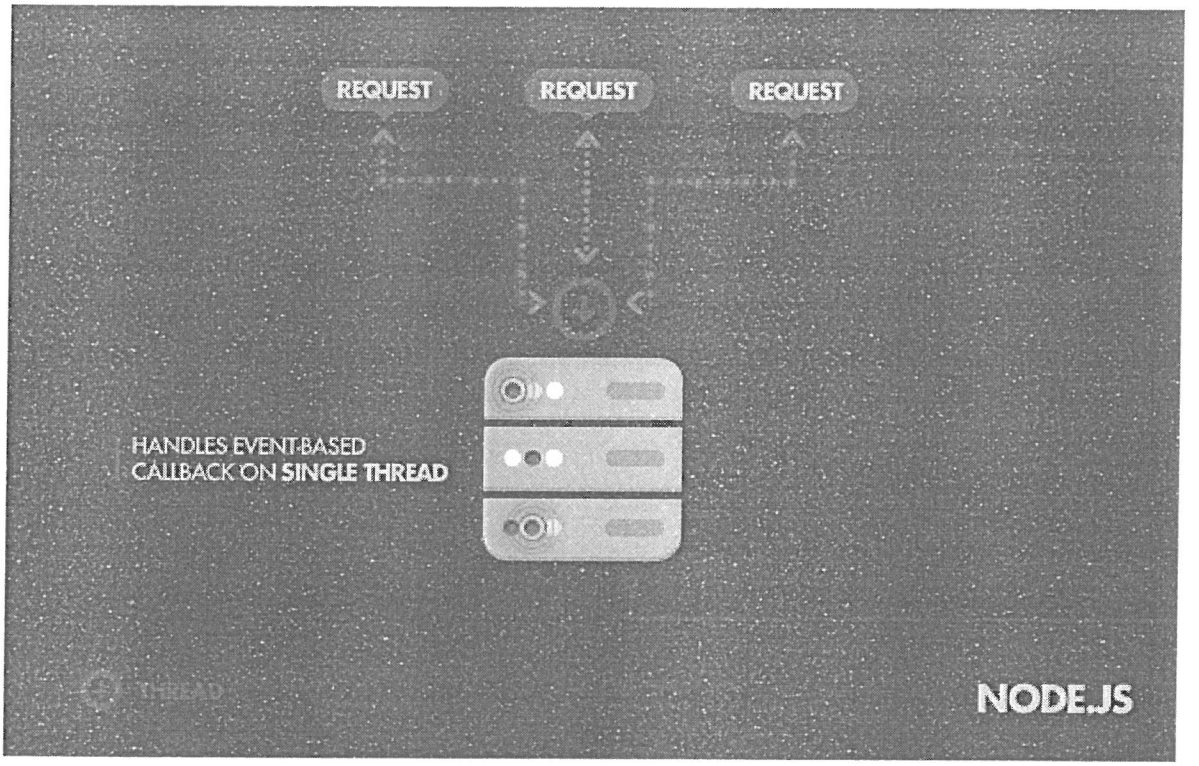


Figure 16. Node js

In JavaScript, we have loops for executing functions, but that itself was not enough for longer executing functions. For example, if we want to execute our process with 60FPS, only 16ms are left to execute the code before the next code frame. JavaScript does not support multi-threading as the JavaScript interpreter in the browser is single-threaded. Even our famous Web browser, Google Chrome, will not let a single JavaScript page run concurrently as it could cause issues in the application.

One single thread handles the entire event cycle. So, JavaScript has introduced a new concept here for the limitation we are facing, i.e., Web Workers. Web Workers can perform long-running tasks without affecting the thread of execution. A worker is an object created using Constructor, which will run a JavaScript file. This JavaScript file contains code which will run in a worker thread; these workers run in another global context which differs from the current window.

Asynchronous JavaScript simply implies the fact that the user will receive the data in a faster manner. This generally took place whenever the user tries to fetch the data from the API (Application Programming Interface) resource or tries to collect the response from the API itself.

To handle the asynchronous data, we use promises or callbacks which helps users easily fetch responses or data from the API with ease. After understanding the basics of Call Stack, some more technical concepts exist like Event Loop, Web API, and Message Queue. All these are not associated with the JavaScript engine, they are actually related to Browser's JavaScript runtime environment. Whenever any synchronous method or DOM events (like setTimeout or promises or callbacks) comes into the existence, it gets executed into the Web API's and after its timer gets completed it is pushed into the message queue from where the event loop further pushes that message into the Call Stack for its execution.

The task which Event Loop executes is basically determining whether the stack is empty. If it is empty, then that particular event loop takes data from the message queue to Call Stack for execution.

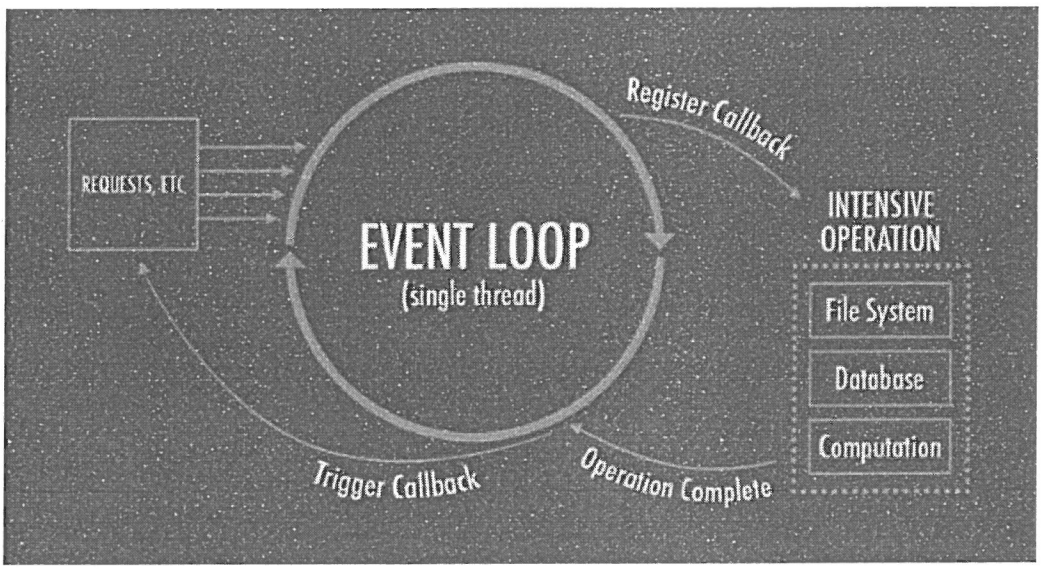


Figure 17. Event Loop Node js

Threading and Scheduling

A thread in Java is the direction or path that is taken while a program is being executed. Generally, all the programs have at least one thread, known as the main thread, that is provided by the JVM or Java Virtual Machine at the starting of the program's execution. At this point, when the main thread is provided, the main() method is invoked by the main thread

A thread is critical in the program because it enables multiple operations to take place within a single method. Each thread in the program often has its own program counter, stack, and local variable.

The Life Cycle of a Thread in Java refers to the state transformations of a thread that begins with its birth and ends with its death. When a thread instance is generated and executed by calling the start() method of the Thread class, the thread enters the runnable state. When the sleep() or wait() methods of the Thread class are called, the thread enters a non-runnable mode.

Thread returns from non-runnable state to runnable state and starts statement execution. The thread dies when it exits the run() process. In Java, these thread state transformations are referred to as the Thread life cycle.

There are basically 4 stages in the lifecycle of a thread, as given below:

1. New
2. Runnable
3. Running
4. Blocked (Non-runnable state)
5. Dead

New State

As we use the Thread class to construct a thread entity, the thread is born and is defined as being in the new state. That is, when a thread is created, it enters a new state, but the start() method on the instance has not yet been invoked.

Runnable State

A thread in the runnable state is prepared to execute the code. When a new thread's start() function is called, it enters a runnable state.

In the runnable environment, the thread is ready for execution and is awaiting the processor's availability (CPU time). That is, the thread has entered the queue (line) of threads waiting for execution.

Running State

Running implies that the processor (CPU) has assigned a time slot to the thread for execution. When a thread from the runnable state is chosen for execution by the thread scheduler, it joins the running state.

In the running state, the processor allots time to the thread for execution and runs its run procedure. This is the state in which the thread directly executes its operations. Only from the runnable state will a thread enter the running state.

Blocked State

When the thread is alive, i.e., the thread class object persists, but it cannot be selected for execution by the scheduler. It is now inactive.

Dead State

When a thread's run() function ends the execution of sentences, it automatically dies or enters the dead state. That is, when a thread exits the run() process, it is terminated or killed. When the stop() function is invoked, a thread will also go dead.

Feature Development

Feature development of add new functionality to the product is also my task during my internship. So, I start with like making the functionality at my local repository then I call it for review and then I just write it in the main code base and then it was published. This is the approach which I have used in the starting after getting the confidence in this, I used to start writing in the main code base with basic testing so that it will cover all the required testcases.

One of my tasks is to create a polling scheduler, the main goal of the task is to provide the user a way to schedule the polling, or you can say schedule the poll after certain interval of time. So, in this task all my learning starting form data structures to algorithm all were included to implement in more effective and function way. Mainly I have used map data structure. And after that the main concepts of java programming used like threading, scheduling, cloning, and copying. The main learning was to use in this that I have to work with past and future time to schedule the API calls. After this I became more effective about handling time with programming, and I also come to know about how to schedule a call after certain interval or time stamp.

There is one more feature on which I have worked upon which was to make excel function more effective. Like I have created support for one more datatype in excel function. For this mainly my work involved in the node js side. Basically, I have created certain function to handle the data from the frontend and then checking their respective types and then finalizing the configuration for the microservice part. This task helps me to understand how we can handle different datatypes in the NodeJS and with the help of different npm modules we can make our functionality more effective to work with. One more thing is that whenever we finalize the high-level design and low-level design of the project we always need to finalize with the US to start the coding process. So, it was also a great learning experience to deal with US peoples or explain them the way we work over the project.

Microsoft Graph API

Microsoft Graph is a Microsoft API developer platform that connects multiple services and devices. Initially released in November 2015 as Office 365 Unified API, the Microsoft Graph builds on Microsoft 365 APIs and allows developers to integrate their services with Microsoft products, including Windows, Microsoft 365, and Azure. At its Build 2017 conference, Microsoft announced it would use the Microsoft Graph to bring new functionality and connectivity between Windows and other OS platforms, including Android and iOS.

The Microsoft Graph API is a RESTful web API that enables you to access Microsoft Cloud service resources. After you register your app and get authentication tokens for a user or service, you can make requests to the Microsoft Graph API.

Microsoft Graph exposes REST APIs and client libraries to access data on the following Microsoft cloud services:

- Microsoft 365 services: Delve, Excel, Microsoft Bookings, Microsoft Teams, OneDrive, OneNote, Outlook/Exchange, Planner, and SharePoint
- Enterprise Mobility and Security services: Advanced Threat Analytics, Advanced Threat Protection, Azure Active Directory, Identity Manager, and Intune
- Windows 10 services: activities, devices, notifications
- Dynamics 365 Business Central

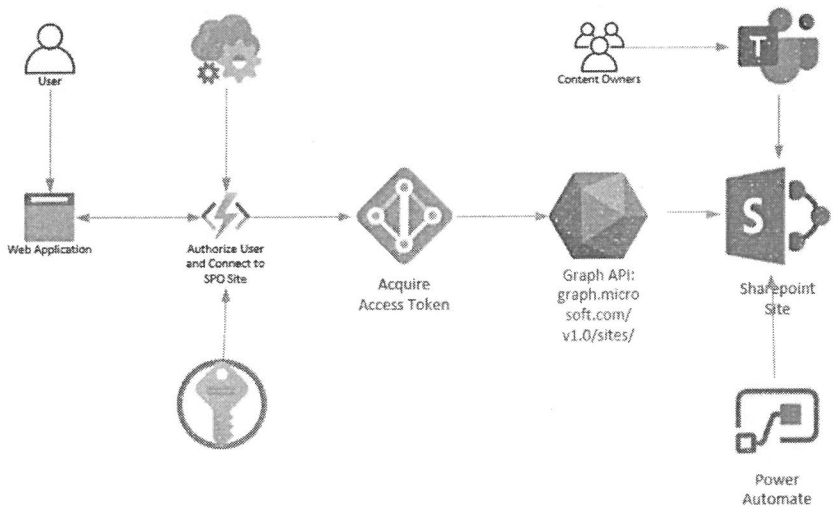


Figure 18. Microsoft Graph API

To use Microsoft Graph to read and write resources on behalf of a user, your app must get an access token from the Microsoft identity platform and attach the token to requests it sends to Microsoft Graph. The exact authentication flow to use to get access tokens will depend on the kind of app you're developing and whether you want to use OpenID Connect to sign the user into your app. One common flow used by native and mobile apps and also by some Web apps is the OAuth 2.0 authorization code grant flow. So basically what we do is that, firstly we register the user to the platform and then we make authentication to access user account with access token and then refresh token issued to make connection active every time the main things come here is that the security of the user, for which we direct connect with the help of access token which is provided by Microsoft in this case so end to end the connectivity and security is maintained.

The graph API is very important while we connect different ecosystem from Microsoft itself. Like it helps us to generate subscription with the help of a REST API call. Through this official graph API, we can directly do API call to make whatever action we want to make with GUI. From creating subscription to add/remove peoples from the channels. The main only things which serve as an input is the notification URL which we have explicitly make in order to acknowledge the API calls.

Therefore, we have created a notification URL in node is so whenever we call the API, it gets acknowledged by the server and then it completes it call. Last thing we have to take care of is the time for which we create the subscription like if we create the subscription of 2 hours then after 2 hour the subscription get lost. Therefore, we also need to take care of the time and while dealing with REST API, I have also learned about tokens which we pass in headers so that we can make a secure API call and also the general methods of rest like get, post, put, patch, delete. So these methods are very helpful and important while doing the API calls.

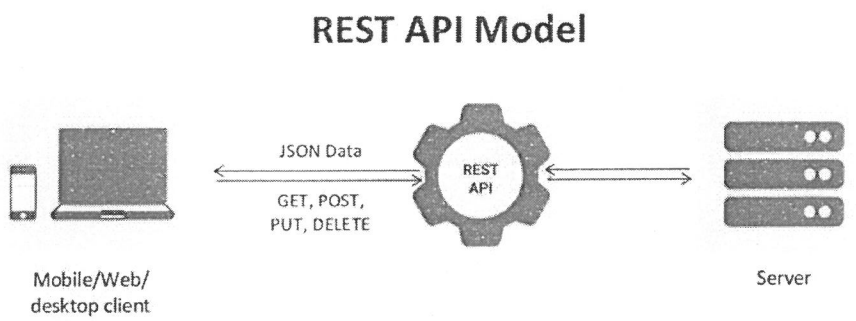


Figure 19. Rest Architecture

As web users progress through a network of web resources each action (such a clicking on a link), causes a change of state. For example, the homepage of a website is one state and when you click on a link you transition to a different state, resulting in the next resource or set of resources, being transferred to you for use.

Representational State Transfer (REST) is an architectural style that defines a specific set of constraints and properties ensuring that its possible for computer systems or software to exchange information. REST is what makes it possible for you to change state easily and continually as you navigate through web resources.

A RESTful API is simply an API that follows the rules of REST specification. REST rules dictate how software components will talk, what kind of message they will send to each other and how requests and responses will be handled. It is an agreed way of doing things.

REST APIs are stateless, meaning that each call for data contains all of the data necessary to complete a call successfully and calls for data are made independently of one another. Using a stateless protocol and standard operations has huge advantages in that REST systems can provide fast performance, reliability and scalability. However, there are also some drawbacks, which fall loosely into two categories: data over-fetching and data under-fetching.

You can make a request from a resource every 10 seconds if you want to and the remote server will (probably) always answer, presuming it hasn't collapsed due to server overload. Instead of calling for data every 10s to find out if any information has changed, you can apply a webhook so that the API will call you when it has new information.

Versioning of the project

Basically, we have our main GitHub repository where we push our changes. So, the main target is the branch on which we are working on like working on some x branch so after then we will create the commit of the work which we have done and then we push the changes to the origin. So here the main concepts of git and GitHub involved like stashing the changes, resolving the merge conflicts, etc.

The workflow

The GitHub workflow can be summarized by the "commit-pull-push".

Commit

- Once you've saved your files, you need to commit them - this means the changes you have made to files in your repo will be saved as a version of the repo, and your changes are now ready to go up on GitHub (the online copy of the repository).

Pull

- Now, before you send your changes to GitHub, you need to pull, i.e. make sure you are completely up to date with the latest version of the online version of the files - other people could have been working on them even if you haven't. You should always pull before you start editing and before you push.

Push

- Once you are up to date, you can push your changes - at this point in time your local copy and the online copy of the files will be the same.

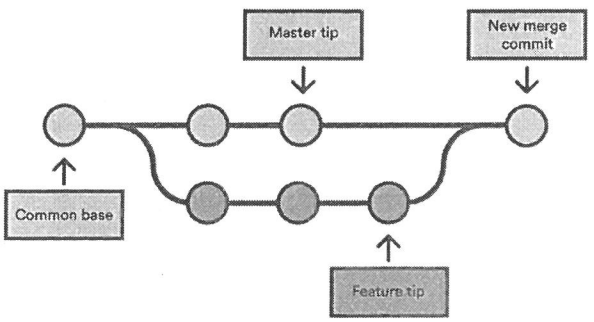


Figure 20. Git Branches

Handling Merge conflicts

A merge conflict is an event that takes place when Git is unable to automatically resolve differences in code between two commits. Git can merge the changes automatically only if the commits are on different lines or branches.

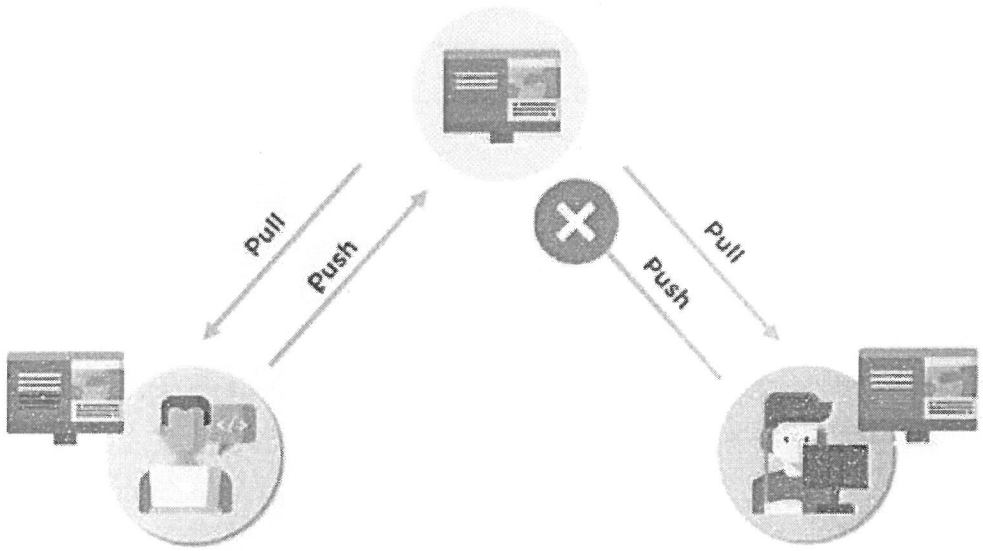


Figure 21. Git Merge Conflicts

Let's take a situation where we have Developer A and Developer B. Both pull the same code file from the remote repository and try to make various amendments in that file. After making the changes, Developer A pushes the file back to the remote repository from his local repository. Now, when Developer B tries to push that file after making the changes from his end, he is unable to do so, as the file has already been changed in the remote repository.

One solution to handle such cases:

1. Use VS Code to handle the code.
2. Just do git status.
3. Then while resolving the conflicts we have some options like accepting the changes, discard the coming changes.
4. So basically, in VS code is become more handy to visualize the changes and directly have an option to accept and reject changes.

Dockerising the Application

Dockerizing is the process of packing, deploying, and running applications using Docker containers. Docker is an open-source tool that ships your application with all the necessary functionalities as one package. You can use Docker to pack your application with everything you need to run the application (such as libraries) and ship it as one package - a container. Containers are created from images that specify their precise contents. Dockerizing is a big hit nowadays. All the big names use it - Google, VMware, or Amazon support it.

There are two parts of Docker:

The Docker Engine - a portable packaging tool

The Docker Hub - cloud service for sharing applications

The Docker container allows another user to quickly recreate the computer environment. The container provides operating virtualization by abstracting the "user space". This technology ensures that your colleagues are developing or testing the product using the same environment as you which results in fewer errors.

Advantages of using docker:

Easy to use Docker simplified the way we deploy applications. You do not distribute the software as source code - you send the binary image of a part of your disk. It is easy to use for developers, system admins, architects, and others. It runs the application unmodified on any cloud. Docker also lets you use the Docker Hub, which contains tens of thousands of public images ready to use.

Fast: Docker containers are just sandboxed environments that run on the kernel. You can create and run the containers in seconds.

Able to create a reproducible environment: Docker creates a consistent work environment that is necessary for recreating applications and their functionalities. Wrapping everything into containers means that the application you build runs on other devices without friction.

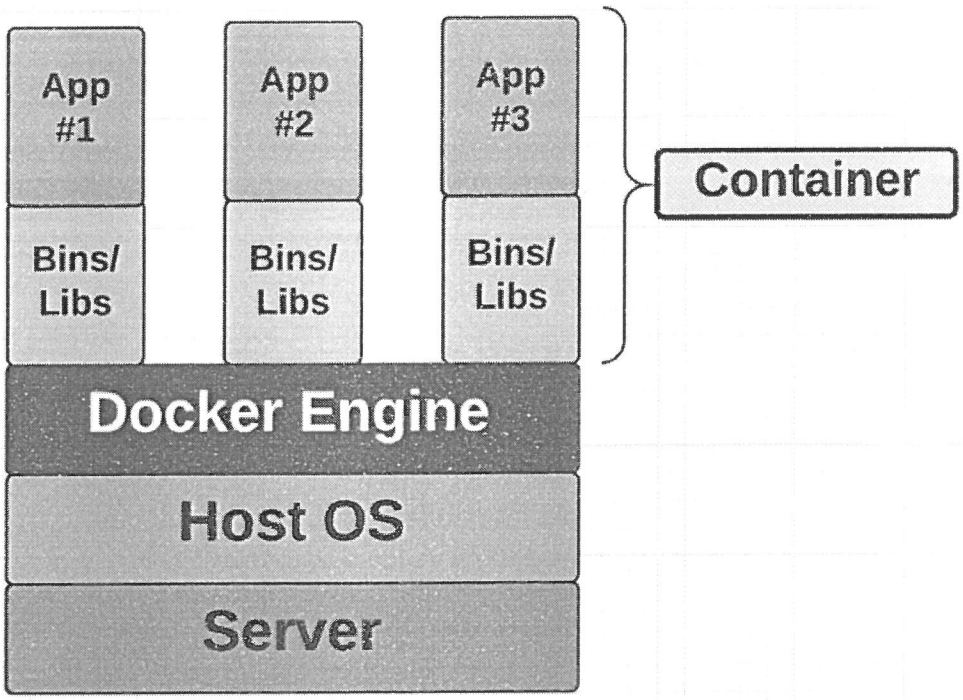


Figure 22. Dockerization

How we dockerise the Application

Steps 1. Create a docker file with the basic configuration.

a) Mentions the base OS Image name. (Alpine in case of node.js)
b) Command to run the service.
c) General pipeline to flow the process to automation.

Step 2. After creating the docker file. Make account on the docker hub. It is basically a platform where we push our docker image. Just like we use GitHub to host our code in the cloud platform.

Step 3. After creating the docker configuration, we have created a container of that particular image, the main things here is binding the important dependencies so that whosoever is going to use the container or pull the container can directly use the project directly. Therefore, we need to take care of the configuration while writing the docker file.

Building docker image with Jenkins

Jenkins Pipeline (or simply "Pipeline") is a suite of plugins which supports implementing and integrating continuous delivery pipelines into Jenkins. A continuous delivery pipeline is an automated expression of your process for getting software from version control right through to your users and customers.

Jenkins Pipeline provides an extensible set of tools for modeling simple-to-complex delivery pipelines "as code". The definition of a Jenkins Pipeline is typically written into a text file (called a Jenkinsfile) which in turn is checked into a project's source control repository.

Steps to work:

1. Create a Jenkins file.
2. Add agent and stages to start.
3. Add pipeline and step in Jenkins file.

After that we will use our Jenkins console so that, we can push new changes to the build of docker. Here we also connect out git repository. It is like a pipeline which track changes so internally it checks for new commits and pushes to the respective branch or the branch on which we build the image. After getting the changes from the git. It starts taking the Jenkins file and load the agent which means the image. And add the new changes to the code of the earlier image. In this way all the changes will be pushed to the earlier version. So, after completion of this process a new tag is applied to the image so that we can maintain the versioning, in future we want to go back we can switch to older version by just changing the tag. In this way versioning work in Jenkins so that we can just apply the tag use the features.

After creation of the docker container we can use Kubernetes to manage and handle the docker containers like for a particular purpose we can have various container. That why we need an efficient tool to handle the docker containers we use Kubernetes to scale up and down and set replicas for the container. Like we just change the tag in the configuration and all the new changes will automatically applied to the new version/tag.

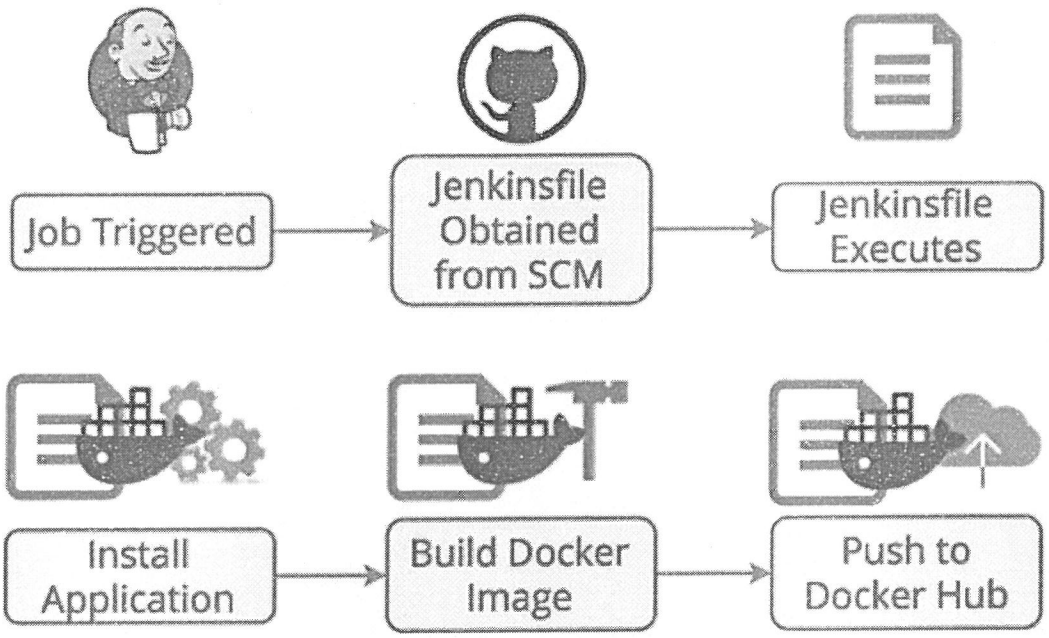


Figure 23. Image Build with Jenkins

Steps to work:

This plugin allows containers to be dynamically provisioned as Jenkins nodes using Docker. It is a Jenkins Cloud plugin for Docker. This docker plugin aims to be able to use a Docker host to dynamically provision a docker container as a Jenkins agent node, let that run a single build, then tear-down that node, without the build process (or Jenkins job definition) requiring any awareness of docker.

As the first step, you need to install the Docker plugin for Jenkins. Whenever we are using docker in our Jenkins pipelines, Jenkins create a "Cloud Agent" using the "Docker" plugin. This agent will be a "Docker Container, "configured to communicate with our Jenkins server's Docker Daemon. The Pipelines with build jobs will use this agent container to execute Docker images to build steps.

Then, the docker image can be pushed into a Docker container registry for deployment. Frist, Head-over to Jenkins server and select "Manage Jenkins", then select the "Manage Plugin" option under "system Configuration."

Now, Click the "Available" tab to view all the Jenkins plugins that can be installed. Search for "Docker" within the search box. Several plugins are available named "Docker" but choose the exact plugin which comes under the "Cloud Providers" heading. Then, click the "Install Without Restart" option and install the plugin.

Once the plugin has been configured, you can define how the Docker plugin needs to launch the Docker containers. This configuration tells the Docker plugin which Docker image to use as the agent. , Docker plugin commands the agent to use a specific Docker image. And after, Docker-Agent Container host the Docker environment to build our Docker image where we defined in the Jenkins pipeline.

When the Docker build request comes from the Jenkins server, Docker-Agent starts to spin up a new container for the Jenkins pipeline task/job. For the sake of simplicity, This Docker-Agent container spins up when the Jenkins build job/pipeline triggered from the Jenkins server. Now, Let's head over to "Manage Jenkins" and select the "Configure System" section. Scroll to the bottom of the page, and there is a section called "Cloud". And click on the "A separate configuration section" link. Manage Jenkins > Configure System > [Scroll Down to Far Bottom] > Cloud > [Configure Clouds]

The Docker Agent Template is the container that will be started to handle the build process.

Let's head-over to Manage Jenkins > Configure System > [Scroll bottom] > Cloud > Configure Cloud. Now, click on "Docker Agent Template" and then "Add Docker Template". Here, you can configure the container options. You can attach these templates when you start building docker containers. I'm using the "jenkins/agent:latest" image for the demonstration purpose, which has the docker client inside. Here, "Labels" and "Name" cam use to associate a job to a particular agent. This means that Jenkins's labels build jobs to indicate that they should be built via the Docker agent that we are running here.

In a Kubernetes ecosystem, for a container to stay alive, it needs to run an image. These images are a snapshot of a system or application's entire state saved up in a non-volatile file format. An image represents how an application, and its dependencies are packaged in binary format. When administering a Kubernetes application, it is necessary to create one or more images and have them stored in a repository where they can be pulled when they need to be referred into a POD.

A registry is an images repository for development, testing, and access control. With a container registry, software teams can perform security and vulnerability analysis on container images, improve functionality then deploy them to Kubernetes clusters. To use an image in a cluster, its name and registry are specified in the POD or Deployment configuration file. If a registry hostname is not provided, Kubernetes assumes the Default Docker Registry.

Different versions of the same image may be needed for different use cases. Tags are added to image names to identify various versions and help isolate the different usage environments. If a tag is not specified in the manifest file, Kubernetes will automatically use the image tagged "latest".

To pull an image from a container registry, the following two steps are required:

1. Creating a Secret that contains the login credentials used to access the registry. PODs with this secret specified can pull images from the specified registry.
2. The image is pulled from the cloud registry by specifying the image name and location in a POD Spec or Deployment YAML file.

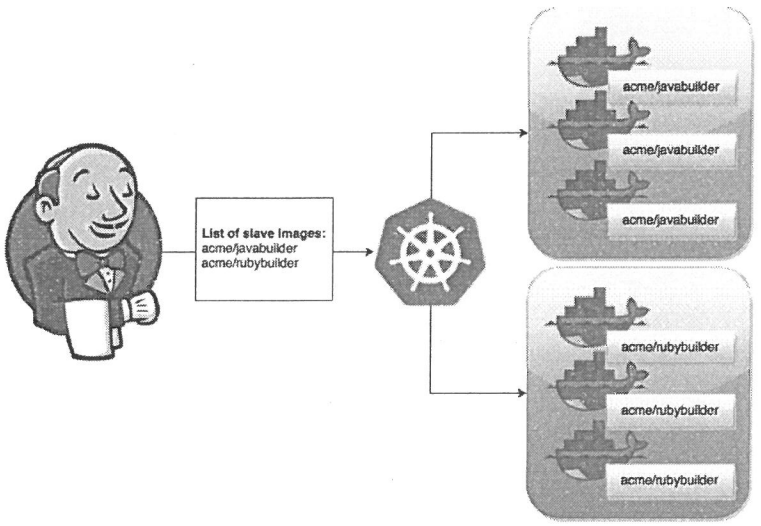


Figure 24. Jenkins and Docker

Website Handling with WordPress

Along with the backend development, with my experience in WordPress development I was also handling the WordPress website. So mainly my work includes fixing the bugs or changing the configuration so that the main website of the product become faster and more efficient. The tasks in the WordPress were on both frontend and backend development. So, I have also worked on elementor or designing the frontend and also the backend side work which is manly related to handling things with the help of plugins.

WordPress

WordPress (WP, WordPress.org) is a free and open-source content management system (CMS) written in PHP and paired with a MySQL or MariaDB database with supported HTTPS. Features include a plugin architecture and a template system, referred to within WordPress as Themes. WordPress was originally created as a blog-publishing system but has evolved to support other web content types including more traditional mailing lists and forums, media galleries, membership sites, learning management systems (LMS) and online stores. One of the most popular content management system solutions in use, WordPress is used by 42.8% of the top 10 million websites as of October 2021.

WordPress was released on May 27, 2003, by its founders, American developer Matt Mullenweg and English developer Mike Little, as a fork of b2/catalog. The software is released under the GPLv2 (or later) license.

To function, WordPress has to be installed on a web server, either part of an Internet hosting service like WordPress.com or a computer running the software package WordPress.org in order to serve as a network host in its own right. A local computer may be used for single-user testing and learning purposes.

Frontend

Elementor is a drag-and-drop page builder for WordPress. This plugin helps you create beautiful pages using a visual editor. It's designed for you to build dynamic websites quickly. This WordPress plugin is an all-in-one solution — letting you control every part of your website design in a single platform. You can customize your website to fit your brand with motion effects, multiple fonts, and enhanced background images. Elementor is an effective tool for building anything — from entire websites to specific landing pages. You can design sales pages, promotional pages, and even custom forms. If you can imagine it, this platform can help you design it.

Cascading Style Sheet

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts.[3] This separation can improve content accessibility; provide more flexibility and control in the specification of presentation characteristics; enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, which reduces complexity and repetition in the structural content; and enable the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.

Along with that the fundamentals of frontend designing are used to make effective and responsive design. Like various pages were designed with the help of some special themes and specific designing is used. Like making a particular section more focused like pages of blogs and pages which consists of videos. After that the optimization also come like making load time faster as possible. For this we can use cloud-based platform to make delivery faster. For example, like making all the static content uploaded in the AWS S3 bucket, so that it will load and deliver faster than usual.

Backend

The backend part of WordPress mainly includes the work of PHP and MYSQL. So basically, some of the pages and logics of particular event are explicitly written in PHP. So, I have started by exploring or debugging the code there are difference parts of website like staging, development or production. We work on the development phase and when we are done with the development then the work is pushed to the production phase and our changes will be live.

In PHP the unnecessary code or buggy code was removed so that there is a chance to improve allover performance and also for improving the experience of the user from all perspective.

PHP

HP code is what makes WordPress work. A WordPress website stores all its data in a MySQL database. This data includes everything from your blog name and blog post content to your plugin settings and user profile information.

PHP's job is to get specific information from the database, and piece it together into an HTML web page. For more details on that process, you can see our guide on how WordPress works behind the scenes. PHP code is contained in files that end in the .php extension. The term "server-side" just means that all those PHP files in WordPress do their work on your web hosting server. So, in simple terms, when someone goes to your website, WordPress accesses the PHP files to get your database information (like your blog post content) to show it to your visitor. PHP is a logic-programming language which you can use to control which HTML a page shows, either in WordPress or outside of it. The files that make up both WordPress themes and plugins are mostly using PHP to build the pages that you see when you visit a WordPress site in your web browser.

Testing of the features and finalizing the test cases

So basically, whenever we are done with the development, we start analyzing the different situation and different test cases for the code changes. For example, like in the document I have explained the features which I have made was polling scheduler so after completion I come to know that there are various testcases where code might get failed. Like at different time the time difference is always different. Then the logic is changed to the specific calculation with respect to the time also there are various edge cases like scheduling the poll on 11:59 and here the time remains only is 1 minutes in the completion of day. That's why whenever we code thinking and analyzing different cases will be more important rather than writing the code directly.

Research on Topics for new implementation

In the internship as a developer, we have also researched on various topics for the new project for implementation. Like I have researched on Graph API for the subscription generation service so that we can generate new subscription for the clients and seamless connection with the user account. Mainly I have researched on shared mailbox where when a user is terminated his/her account access is given to the administrator. For this research will be on permission of the mailbox is setup so that we can access the account.

Other than shared mailbox, I have also researched on the Workday. Workday is basically a HR system. The research is on how we can connect with workday, is there any mechanism to connect with workday, what is the authentication used for the workday. So as a result, I have listed down all the possible ways to connect with workday using rest API, there is procedure to connect with workday, like we have created access token and refresh token in order to access the data.

Summary

As a part of Backend Development Team my learning was immense, and I have learned a lot in the field of development. Till my college I have learned and for implementing my learning I just made projects. But working with real project was a different experience I came to know how we have made changes according to the client and how to make logics under constraints so that result product will be effective and fast. My main projects were:

1) Connect iPass.
2) Hire 2 Retire.

My main technology includes Java, Java Spring, Microservices, Nodejs and express. As these technologies are mostly used to make backend related services, therefore learning the in-depth concepts like creating middleware, authentication for application was also very interesting. There are lots of resources out there which helped me to learn and implement the backend related task.

Following the modular approach, abstraction and functional based code flow is important part of the writing the code. Like most of the fundamental which we used to follow like encapsulation, polymorphism using these in real time for example creating the interface and implementing that interface in other projects so that it will be act as a base class and the actual functionality is remain as it ease.

Along with that advanced concept of java like threading or task scheduling has been used to handle time taking task like scheduling a task after x minutes, here thread was involved so that we can process task parallel manner and faster manner.

References

1. https://www.robomq.io/
2. https://www.javatpoint.com/java-tutorial
3. https://www.rabbitmq.com/tutorials/tutorial-one-python.html
4. https://git-scm.com/docs/gittutorial
5. https://docs.github.com/en/get-started/quickstart/hello-world
6. https://www.javatpoint.com/spring-tutorial
7. https://www.tutorialspoint.com/junit/index.htm
8. https://www.w3schools.com/sql/
9. https://docs.docker.com/get-started/
10. https://kubernetes.io/docs/tutorials/
11. https://www.postgresqltutorial.com/
12. https://www.javatpoint.com/linux-tutorial
13. https://docs.docker.com/
14. https://kubernetes.io/
15. https://www.jenkins.io/doc/
16. https://microservices.io/
17. https://nodejs.org/en/
18. https://www.rabbitmq.com/tutorials/tutorial-one-javascript.html
19. https://developer.wordpress.com/docs/
20. https://developer.mozilla.org/en-US/
21. https://www.javascript.com/resources

DocuSign Envelope ID: 9EA65633-E7ED-409B-9A9A-5232C6249FF7

GST No. **08AAICR2941J1Z4**
Mob no. +91-9829013033
Email: Mohanlal.goel@robomq.io
mohanlalgoel@gmail.com

RoboMQ Pvt. Ltd.

Headquarters
8260 Greensboro Dr.,
Ste A-32, McLean, VA 22102
USA

India Center
307 Sankalp Tower, 8 Queens Rd,
Jaipur, Rajasthan 302021
India

To Whom-So-Ever It May Concern

Respected Ma'am/Sir,

This is to certify that Harsh Udai of JK Lakshmipat University has completed his internship from January 3rd 2022 to June 23rd 2022 under the supervision of our product leads on our product **RoboMQ- Connect iPaaS and Hire2Retire.**

He has completed his training successfully and his performance was good.
We wish him a bright future ahead.

DocuSigned by:
Prachi Sharma
335A8C28C90444A...

Date: 7/7/2022

Name: Prachi Sharma

Prachi Sharma
HR Manager and Executive Assistant
Prachi.sharma@robomq.io
+91-8000844735