**PROJECT REPORT**

**DevOps Assignment**

**Team Members:-**

* Palash Das – 2020HS70002
* Deepika Sharma – 2020HS70016
* Renu Rachael Johnson – 2020HS70049
* Naima Ismail – 2020HS70034
* Anil Krishnan – 2020HS70023

**Title of the Project**

**CoSAP (Vaccine Management System for SAP Labs) -** Web Application Implementation with Microservices

**Define Git/GitHub Workflow**

[Repository Link](https://github.com/dpalash608/CoSap-Java)

**Show evidence of having followed the same**

**Steps to run the application**

1. Clone the repository mentioned above.
2. Import it in any of the suitable IDE (VSCode/Eclispe).
3. Run mvn clean install on the project.
4. In the main application file cosapapplication.java, run this file as a JAVA application.
5. In the console, you will see the server getting started.
6. Now paste the below content in the json file and import it in postman.
7. You will see in the imported collection set of API’s which you can use.

{

"info":{

"\_postman\_id":"e2a9f91f-396c-45dd-8e29-81931897f2fc",

"name":"cosap",

"schema":"https://schema.getpostman.com/json/collection/v2.1.0/collection.json"

},

"item":[

{

"name":"api/v1/employee/i516845",

"request":{

"method":"GET",

"header":[

],

"url":{

"raw":"http://localhost:8080/api/v1/employee/i516845",

"protocol":"http",

"host":[

"localhost"

],

"port":"8080",

"path":[

"api",

"v1",

"employee",

"i516845"

]

}

},

"response":[

]

},

{

"name":"api/v1/employee?id=516845",

"request":{

"method":"GET",

"header":[

],

"url":{

"raw":"http://localhost:8080/api/v1/employee?id=i516845",

"protocol":"http",

"host":[

"localhost"

],

"port":"8080",

"path":[

"api",

"v1",

"employee"

],

"query":[

{

"key":"id",

"value":"i516845"

}

]

}

},

"response":[

]

},

{

"name":"api/v1/employee",

"request":{

"method":"GET",

"header":[

],

"url":{

"raw":"http://localhost:8080/api/v1/employee",

"protocol":"http",

"host":[

"localhost"

],

"port":"8080",

"path":[

"api",

"v1",

"employee"

]

}

},

"response":[

]

},

{

"name":"api/v1/employee/i516845",

"request":{

"method":"DELETE",

"header":[

],

"url":{

"raw":"http://localhost:8080/api/v1/employee/i516845",

"protocol":"http",

"host":[

"localhost"

],

"port":"8080",

"path":[

"api",

"v1",

"employee",

"i516845"

]

}

},

"response":[

]

},

{

"name":"api/v1/employee/i516845",

"request":{

"method":"PUT",

"header":[

],

"body":{

"mode":"raw",

"raw":"{\r\n \"iNumber\":\"i516845\",\r\n \"empName\":\"Palh\",\r\n \"empMail\":\"pal@gmail.com\",\r\n \"empContact\":\"7890282236\",\r\n \"password\":\"fhhfch\"\r\n}\r\n"

},

"url":{

"raw":"http://localhost:8080/api/v1/employee/i516845",

"protocol":"http",

"host":[

"localhost"

],

"port":"8080",

"path":[

"api",

"v1",

"employee",

"i516845"

]

}

},

"response":[

]

},

{

"name":"api/v1/employee/all",

"request":{

"method":"POST",

"header":[

],

"body":{

"mode":"raw",

"raw":"[\r\n {\r\n \"iNumber\":\"i516845\",\r\n \"empName\":\"Palh\",\r\n \"empMail\":\"pal@gmail.com\",\r\n \"empContact\":\"7890282236\",\r\n \"password\":\"fhhfch\"\r\n },\r\n {\r\n \"iNumber\":\"i516245\",\r\n \"empName\":\"Pasd\",\r\n \"empMail\":\"pasd@gmail.com\",\r\n \"empContact\":\"7892282236\",\r\n \"password\":\"fh23ch\"\r\n }\r\n\r\n]\r\n",

"options":{

"raw":{

"language":"json"

}

}

},

"url":{

"raw":"https://localhost:8080/api/v1/employee/all",

"protocol":"https",

"host":[

"localhost"

],

"port":"8080",

"path":[

"api",

"v1",

"employee",

"all"

]

}

},

"response":[

]

},

{

"name":"api/v1/employee",

"request":{

"method":"POST",

"header":[

],

"body":{

"mode":"raw",

"raw":"{\r\n \"iNumber\":\"i516845\",\r\n \"empName\":\"Palh\",\r\n \"empMail\":\"pal@gmail.com\",\r\n \"empContact\":\"7890282236\",\r\n \"password\":\"fhhfch\"\r\n}\r\n",

"options":{

"raw":{

"language":"json"

}

}

},

"url":{

"raw":"https://localhost:8080/api/v1/employee",

"protocol":"https",

"host":[

"localhost"

],

"port":"8080",

"path":[

"api",

"v1",

"employee"

]

}

},

"response":[

]

}

]

}

**Problem Statement:**

The following problems due to which our application will provide solutions:

* Management of Vaccine for all the employees within SAP.
* Maintaining data and records of both the doses.
* Continuous integration and deployment.

**Why is the particular topic chosen?**

**Web Application – Co-SAP (Vaccine Management System for SAP Labs)**

Considering the need of such a system in the current scenario of the pandemic, the topic is chosen to resolve the above problem to help increase the vaccine drive.

**Features:**

* Register with I number (unique identification number for employees of SAP).
* Book vaccine – option for both dose 1 and dose 2.
* Add, delete, and update status reports.

**Hardware and Software Used**

**Developer End**

* Intel Core i7
* 4GB Ram
* Windows 7 and above
* Languages- Spring Boot
* Visual Studio Code
* H2 embedded database

**Take a simple use case requiring scalability**

**Challenges faced**

**Inadequate risk management**

Having the foresight to identify potential ‘what if’ scenarios and making up contingency plans is an important aspect of project management. Projects rarely go exactly as planned because there are so many variables that can create unlimited possibilities.

**How we dealt with it:**

It is the job of every project manager to come up with alternate plans that the team may adopt if the project begins to spiral out of control. Having a [project risk management](https://kissflow.com/project/project-risk-management/) system helps in identifying the types of risks and mitigating them. Having a contingency plan in place is critical. This plan should identify all risks that the course of action to be taken if they materialize.

So, we identified the risks and simplified them.

**Integrated environment for the project**

* IDE – Eclipse / Visual Studio …
* Git
* GitHub
* Maven/Gradle
* SonarQube
* Selenium

[Git](https://git-scm.com/about)

Git is a [DevOps tool](https://www.simplilearn.com/tutorials/devops-tutorial/devops-tools) used for source code management. It is a free and open-source version control system used to handle small to very large projects efficiently. Git is used to tracking changes in the source code, enabling multiple developers to work together on non-linear development. Git is used to tracking changes in the source code

* The distributed version control tool is used for source code management
* It allows multiple developers to work together
* It supports non-linear development through its thousands of parallel branches

[GitHub](https://github.com/about)

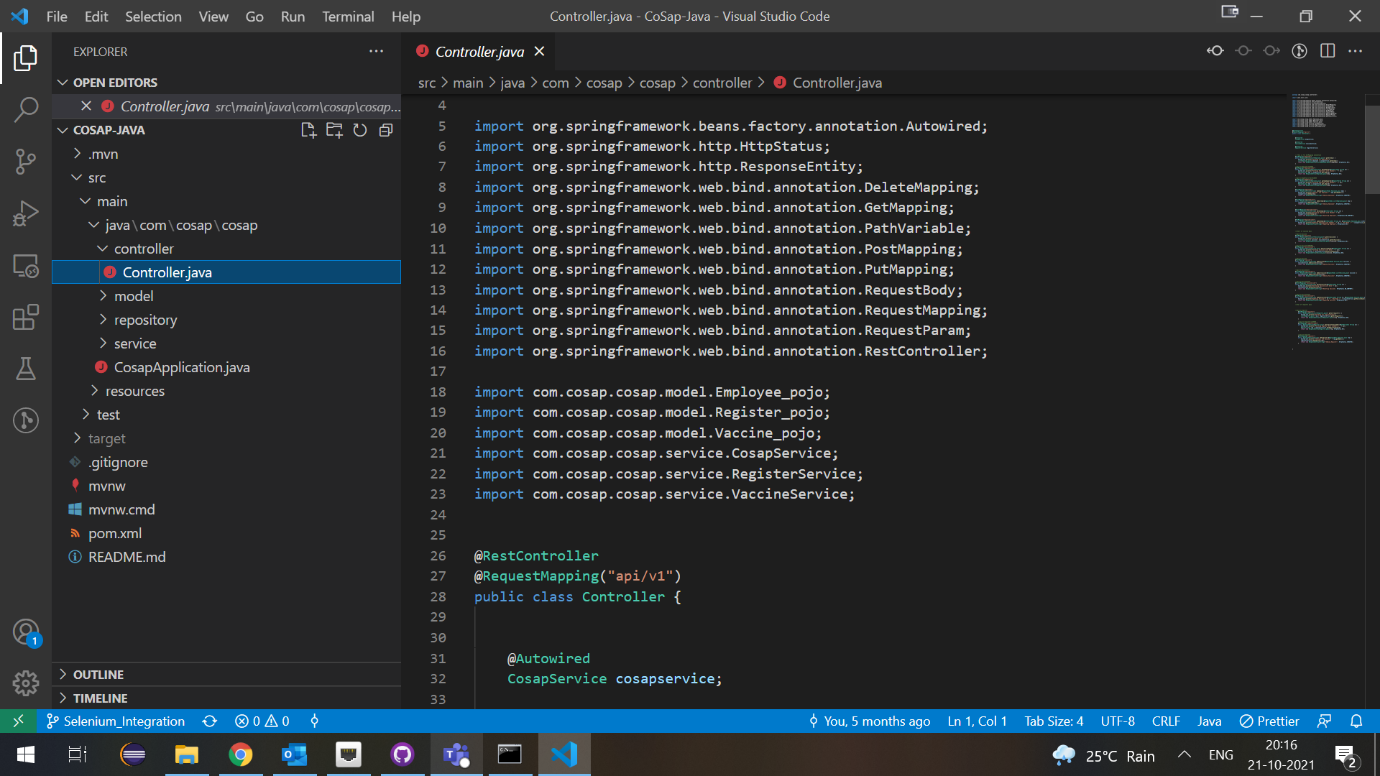
At its core, GitHub is a platform where hundreds of millions of private, public, and open source repositories are hosted and reviewed. Not only is GitHub is the #1 hosting service in our DevOps report, it came in at #7 in our report of the Top 20 Developer Tools for 2020.

GitHub is increasingly expanding its offerings to align with more and more processes in the DevOps workflow. To interface with GitHub repositories, many developers use the GitKraken Git GUI, which seamlessly integrates with GitHub.com and GitHub Enterprise. GitHub offers basic project management with Projects and Issues. View, edit, and create branches tied to these issues directly from GitKraken through the GUI’s robust issue tracking integration.

[VS Code](https://code.visualstudio.com/)

VS Code is a very popular code editor for writing, building and debugging web and cloud applications on Windows, Mac and Linux.

As a Microsoft tool, it has the added advantage of tight integration with Azure, AWS, .NET and a vast ecosystem of extensions which allow you to connect, build and debug many tools and technologies. Streamline your DevOps workflow by using VS Code with Azure to easily deploy and host sites built on React, Angular, Vue, Node, Python, etc.



[Maven](https://maven.apache.org/)

Maven is a build automation tool used primarily for Java projects but can also be used to build and manage projects written in C#, Ruby, Scala, and other languages. The Maven project is hosted by the Apache Software Foundation.

Teams can use Maven’s project object model (POM) and set of plugins to build projects with a unified build system. Once your team is familiar with how one Maven project builds, you’ll know how all Maven projects build, saving time when trying to navigate numerous projects.

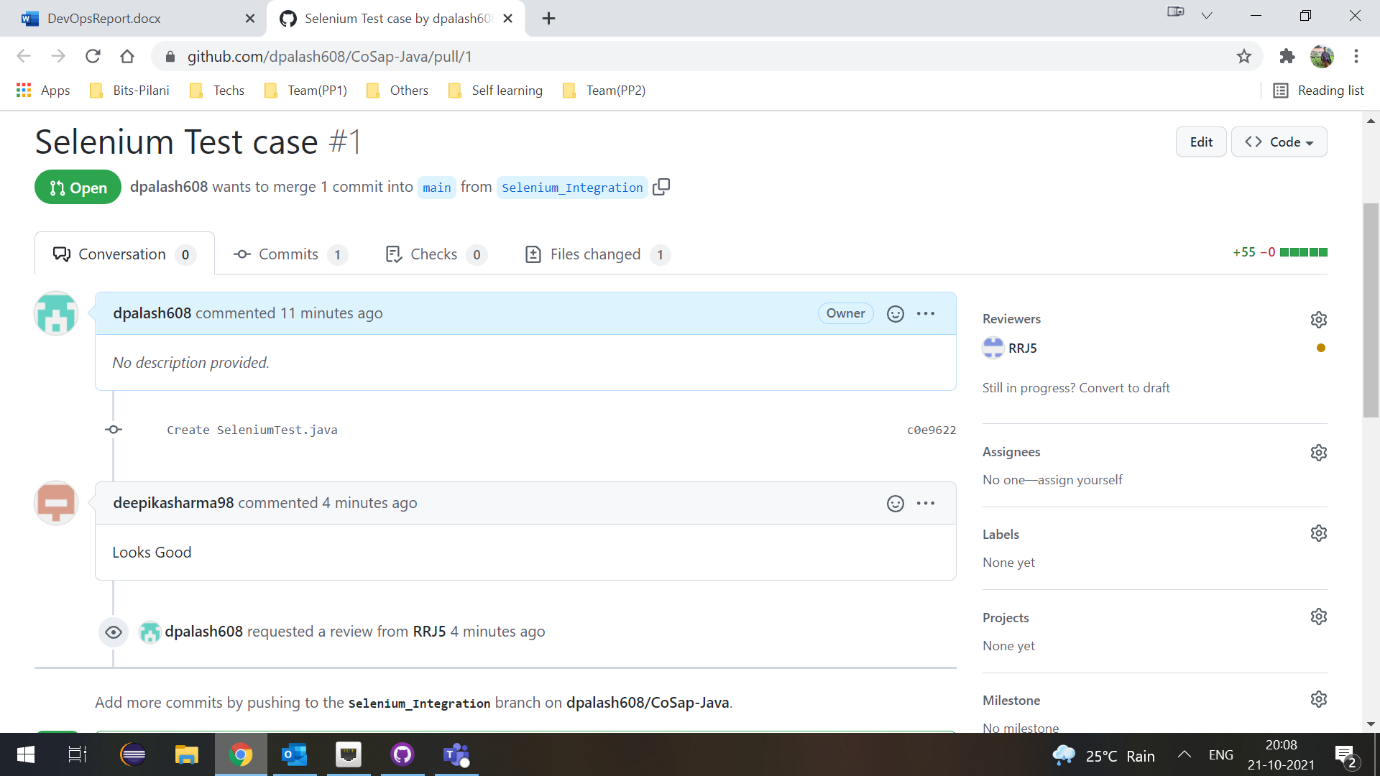
[Selenium](https://www.selenium.dev/" \t "_blank)

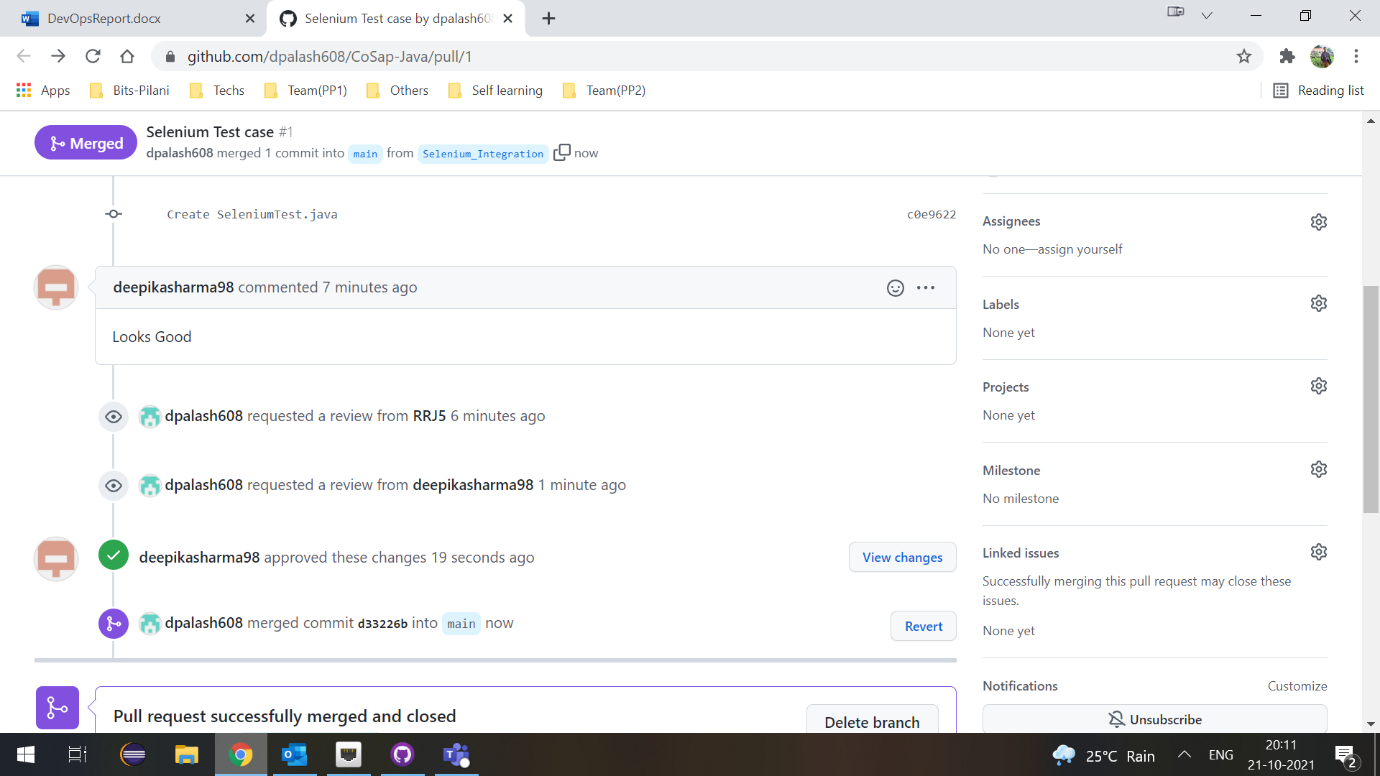
Selenium is a suite of tools for automating web browsers. It provides a playback tool for authoring functional tests without the need to learn a test scripting language.

Selenium WebDriver is a collection of language-specific bindings to drive a browser. It helps QA teams create robust, browser-based regression automation suites and tests, and scale/distribute scripts across many environments.

Selenium IDE is a Chrome and Firefox add-on that will do simple record-and-playback of interactions with a browser. It helps QA teams create quick bug reproduction scripts and scripts for automation-aided exploratory testing.

Selenium Grid is ideal for QA teams who want to scale by distributing and running tests on several machines while managing multiple environments from a central point. This makes it easy to run tests against a variety of browsers and operating systems.





[SonarQube](https://www.sonarqube.org/about/)

SonarQube is the leading tool for continuously inspecting the Code Quality and Security of your codebases and guiding development teams during Code Reviews.

Fully integrated with DevOps tool chains it comes with:

* Built-in integration with most build tools, which enables in most cases a no configuration approach
* Easy integration with continuous integration engines such as Jenkins, Azure DevOps, TeamCity, Bamboo.
* Support for numerous source configuration management tools such as Git, Subversion, CVS, Mercurial.

