

# **International Institute of Information Technology, Bangalore**

## **SOFTWARE PRODUCTION ENGINEERING**

### **Mini Project**

Project report on

***Code IITB - A Coding Platform and Discussion Forum***

**In the Guidance of: Prof. B. Thangaraju**

**Teaching Assistant: Harsh Tripathi**



**Developed by :**

Nisarg Shah MT2019071

Rachit Ban

MT2019083

Nirmal Patel MT2019070

Aayushya Vadher

MT2019003

# CONTENTS

<b>1</b>	<b>Abstract</b>	<hr/> 6
<b>2</b>	<b>Introduction</b>	<hr/> 7
<b>3</b>	<b>Experimental Setup</b>	<hr/> 8
<b>4</b>	<b>System Configuration</b>	<hr/> 10
<b>5</b>	<b>Software Development Life Cycle (SDLC)</b>	<hr/> 11
<b>5.1</b>	<b>How to run the Project</b>	<hr/> 11
<b>5.2</b>	<b>Source Code Management(SCM)</b>	<hr/> 12
<b>5.3</b>	<b>Build automation</b>	<hr/> 13
<b>5.4</b>	<b>Test</b>	<hr/> 15
<b>5.5</b>	<b>Artifact</b>	<hr/> 16
<b>5.6</b>	<b>Deploy</b>	<hr/> 17
<b>5.7</b>	<b>Monitoring</b>	<hr/> 19
<b>6</b>	<b>The CI/CD Pipeline using Jenkins</b>	<hr/> 23
<b>6.1</b>	<b>Creating Jenkins Pipeline</b>	<hr/> 25
<b>6.2</b>	<b>The Pipeline Phases</b>	<hr/> 30
<b>6.3</b>	<b>Rundeck Integration in Jenkins</b>	<hr/> 31
<b>6.4</b>	<b>Logstash Integration with Jenkins</b>	<hr/> 32
<b>7</b>	<b>Result And Discussion</b>	<hr/> 33
<b>8</b>	<b>Scope For Future Work</b>	<hr/> 41

<b>9</b>	<b>Conclusion</b>	<hr/>	42
<b>10</b>	<b>References</b>	<hr/>	43

## LIST OF SCREENSHOTS

SS No	Screenshot Name	Pg.No.
5.2.1	<i>Project Repository</i>	12
5.3.1	<i>WAR packaging of the project</i>	13
5.3.2	<i>Successful build of Web Archive(WAR)</i>	14
5.4.1	<i>Dependency for JUnit framework</i>	15
5.4.2	<i>Unit test example code</i>	15
5.5.1	<i>Docker Image of the project on DockerHub</i>	16
5.5.2	<i>Dockerfile</i>	16
5.6.1	<i>Rundeck Job</i>	17
5.6.2	<i>Successful Run of Rundeck Job</i>	18
5.7.1	<i>Logstash configuration in Jenkins</i>	19
5.7.2	<i>Timestamp view of logs</i>	20
5.7.3	<i>Creating an Index</i>	20
5.7.4	<i>Creating an index pattern</i>	21
5.7.5	<i>Index configuration setting</i>	21
5.7.6	<i>Visualization</i>	22
6.1.1	<i>Providing github url for automatic checkout</i>	25
6.1.2	<i>Configuring the pipeline build trigger</i>	26
6.1.3	<i>Providing location of pipeline script</i>	27

6.1.4	<i>Pipeline script</i>	29
6.1.5	<i>Pipeline Stage View</i>	29
6.3.1	<i>Rundeck Integration</i>	31
6.4.1	<i>Logstash Integration</i>	32
7.1	<i>Login Page</i>	33
7.2	<i>Registration Page</i>	33
7.3	<i>Problem List</i>	34
7.4	<i>The IDE and Problem Page</i>	34
7.5	<i>Code in IDE</i>	35
7.6	<i>Testing the Code</i>	35
7.7	<i>Correct Output</i>	36
7.8	<i>Compilation Error</i>	36
7.9	<i>Wrong Answer</i>	37
7.10	<i>Editorial Page</i>	37
7.11	<i>Upcoming Contests</i>	38
7.12	<i>Leaderboard Page</i>	38
7.13	<i>Discussion Page</i>	39
7.14	<i>Create Problem Page</i>	39
7.15	<i>Host Contest Page</i>	40

## LIST OF FIGURES

<i>Sr. No.</i>	<i>Figure Name</i>	<i>Pg. No</i>
3.1	<i>Architecture and Workflow</i>	8
3.2	<i>User and System Interaction</i>	9
6.1	<i>The CI/CD pipeline of the project</i>	23

## LIST OF TABLES

<i>Sr. No.</i>	<i>Table Name</i>	<i>Pg. No.</i>
4.1	<i>System Specifications</i>	10
4.2	<i>DevOps Tools</i>	10
6.2	<i>Pipeline Phases</i>	30

## **1. ABSTRACT**

“Code IIITB” is an online coding platform developed for students specific to IIITB. It facilitates an online Integrated Development Environment(IDE) to write code, compile and see the output. With application, students can practice for coding competitions. It can also serve as a test platform for placements. It contains a simple discussion forum. The students can post questions, provide explanations or solutions to problems. The participating students can view where they stand according to the points they gain, by solving problems, on the leaderboard section. We developed the project using the DevOps philosophy so these individual modules can continuously be delivered to users and feedback can be received to improve the user experience of the application. In future any new feature can be inserted with minimal effort from developing side and fast delivery of the application features can be made to the users.

## **2. INTRODUCTION**

Being a computer science student, one needs to develop problem solving skills so as and when he/she works for real world projects, it helps, take on the problem and solve it. Improving upon problem solving needs practising data structure and algorithm problems. Thus our project provides a platform to practice questions online and improve the problem solving skill sets

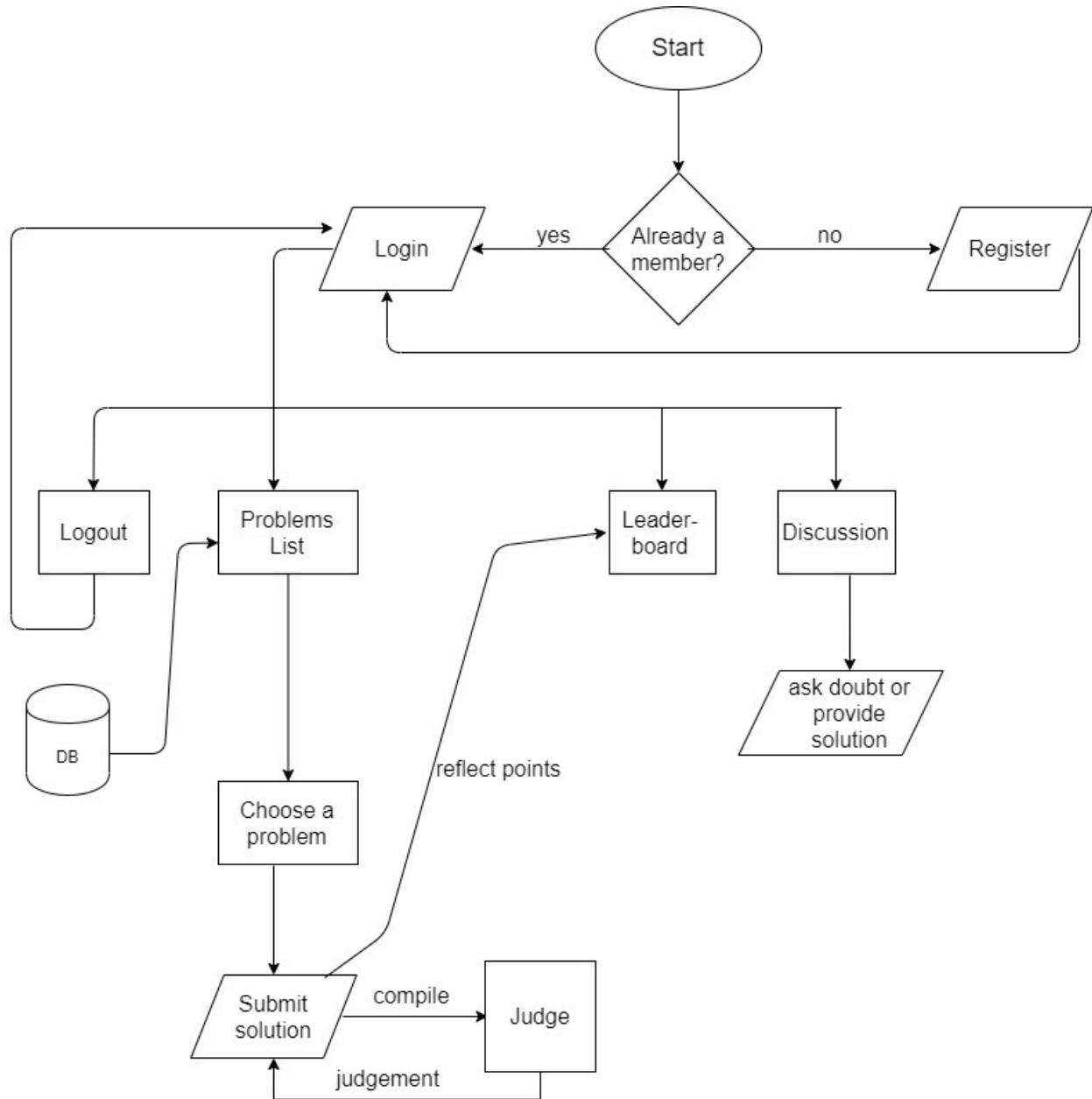
### *The need of DevOps in our project*

DevOps is part of almost every discussion in the project team these days. There is an urgent need for a huge change in the way development and operations teams used to collaborate and communicate earlier. The end objective of the DevOps culture is continuous improvement which leads to achieving new heights with continuous innovations.

Our project includes the modules that are independent of each other ex: solving a problem is independent of creating a contest which means we can start releasing the basic functionalities first and then using DevOps philosophy to continuously release the new modules and features. Continuous Integration (CI), Continuous Testing (CT), and Continuous Delivery (CD) are significant parts of DevOps culture. CI includes automating builds, unit tests, and packaging processes while CD is concerned with the application delivery pipeline across different environments. CI and CD accelerate the application development process through automation across different phases, such as build, test, and code analysis, and enable users achieve end-to-end automation in the application delivery lifecycle.

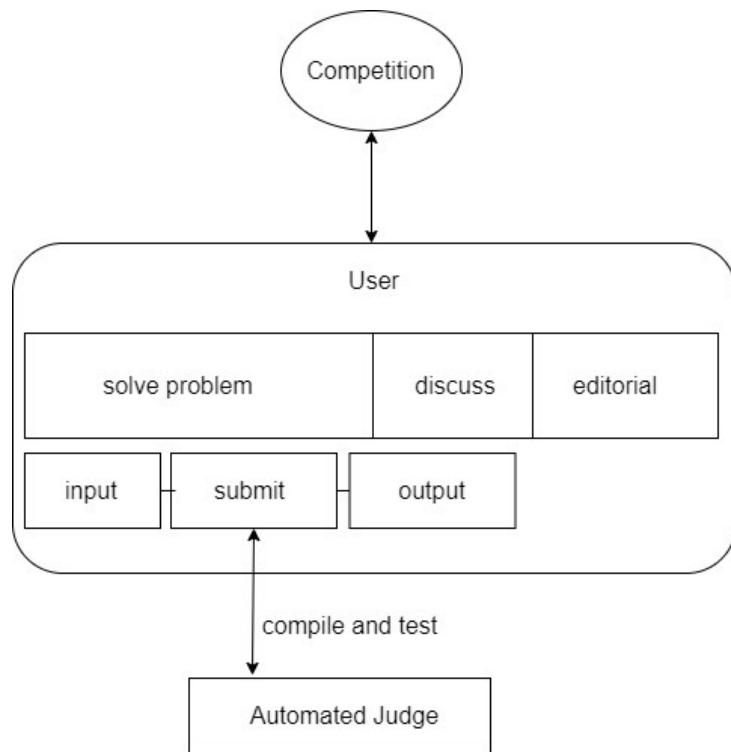
### 3. EXPERIMENTAL SETUP

The basic modules are Login/Register, List Problems, Solve the Problem, Discussion forum and Leaderboard. The flow of the system is shown in *Figure 3.1*



*Figure 3.1 Architecture and Workflow*

*Figure 3.2* shows how the interaction of users takes place with the system. Users have access to the online IDE where he/she can write code, submit (compile), and receive output accordingly. The submitted code is compiled by the Judge of the system and accordingly outputs to the output tab.



*Figure 3.2 User and System Interaction*

## 4. SYSTEM CONFIGURATION

System Specifications	
Machine	Ubuntu 18.04 LTS
Kernel Version	5.3.0-51-generic
RAM	8GB
Web Framework	Spring Boot and JSP [1]
Database service	Aws RDS - MySql [2]

*Table 4.1 System Specifications*

DevOps Tools	
SCM	GitHub
Continuous Integration	Jenkins
Continuous Deployment	Docker
Continuous Delivery	Rundeck
Continuous Monitoring	ELK(elasticsearch, logstash and kibana) stack

*Table 4.2 DevOps Tools*

## **5. SOFTWARE DEVELOPMENT LIFE CYCLE**

### **5.1 How to run the Project**

We will deploy our application as a Docker Container. All the dependencies and packages, that the application requires, will already be installed in the Docker Container hence to run the project, we need to download and run the Docker Image that is hosted at DockerHub

**DockerHub Url : <https://hub.docker.com/r/nisarg20/speproject>**

**Command to run the project :**

**\$ docker pull nisarg20/speproject:spe**

**\$ docker run -p 8085:8085 nisarg20/speproject:spe**

## 5.2 Source Code Management(SCM)

SCM is all about the core project files containing source code, and how these shared files are managed. Source Control Management is a vital system that enables developers to work together on the same development project whether they are in the same room or a continent away. If code gets messy, then SCM allows you to revert to an earlier clean version, which is often the quickest solution to fixing the problem. Some of the well known SCM are:

- Git
- Subversion
- CVS
- BitKeeper
- Rational Clear Case

We have used **Git** as our SCM tool and our code repository is hosted at **GitHub**.

**GitHub Repository Link :** <https://github.com/Nisargshah20/SPEMiniProject>

No description, website, or topics provided.

Manage topics

35 commits 1 branch 0 packages 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

File / Action	Description	Time Ago
.idea	Compiler Added	4 days ago
.settings	First Comm	10 days ago
bin	Home Page Completed and LeaderBoard Added	6 days ago
src	jsp pages updated	1 hour ago
target	Admin added	2 hours ago
.classpath	First Commit	10 days ago
.project	First Commit	10 days ago
Dockerfile	Docker File updated	4 days ago
Jenkinsfile	Jenkins File updated	1 hour ago
a.out	Admin added	2 hours ago
pom.xml	Compiler Added	4 days ago
tc	Admin added	2 hours ago
test.sh	compiler added	4 days ago

*SS 5.2.1 Project Repository*

## 5.3 Build Automation

An automated build helps us create an application build. Build automation is essential for continuous integration and the rest of the automation is effective only if the build process is automated. Apache Ant and Apache Maven are two popular build automation tools.

We use Apache Maven as a build tool in our project. It manages evering using an xml file called POM.xml. A Project Object Model (POM) XML file contains information about the name of the application, owner information, how the application distribution file can be created, and how dependencies can be managed

Features of Apache Maven:

- Easy build process
- Extensible-plugins in Java/scripting language
- Forces a standard directory structure
- Uniform build system

Our project is a web application so we package our project as a WAR(Web Archive) file. It contains all the contents of the web application like: html, css, java scripts, xml, jsp and images.

The SS 5.3.1 shows how our web application can be packaged as WAR file

```
<packaging>war</packaging>
<build>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
        </plugin>
        <plugin>
            <inherited>true</inherited>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-compiler-plugin</artifactId>
        </plugin>
    </plugins>
    <finalName>SPEProject</finalName>
</build>
```

SS 5.3.1 WAR packaging of the project

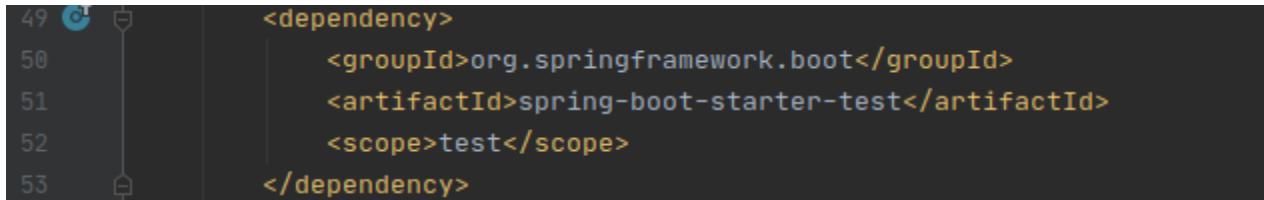
**SS 5.3.2** Shows the successful build of the WAR package after passing all the testcases. We build this using the command: **mvn package**, in the Jenkins pipeline script.

```
[INFO] --- maven-war-plugin:3.2.3:war (default-war) @ SPEProject ---
[INFO] Packaging webapp
[INFO] Assembling webapp [SPEProject] in [/home/nisarg/IdeaProjects/SPEProject/target/SPEProject]
[INFO] Processing war project
[INFO] Copying webapp resources [/home/nisarg/IdeaProjects/SPEProject/src/main/webapp]
[INFO] Webapp assembled in [4272 ms]
[INFO] Building war: /home/nisarg/IdeaProjects/SPEProject/target/SPEProject.war
[INFO]
[INFO] --- spring-boot-maven-plugin:2.2.7.RELEASE:repackage (repackage) @ SPEProject ---
[INFO] Replacing main artifact with repackaged archive
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 59.289 s
[INFO] Finished at: 2020-05-29T12:38:46+05:30
[INFO] -----
```

*SS 5.3.2 Successful build of Web Archive(WAR)*

## 5.4 Test

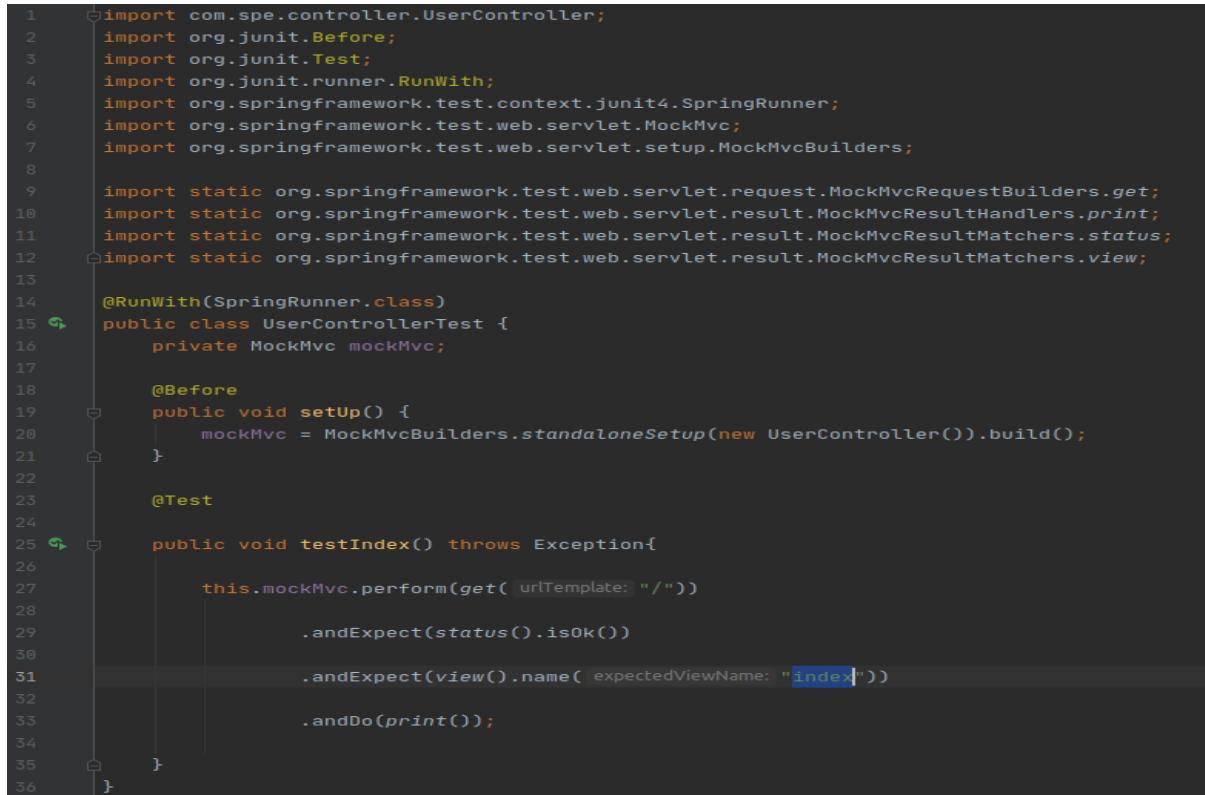
We use **JUnit** framework for unit testing. As test driven development is crucial for development of bug free applications, Java programming language provides a unit testing framework **JUnit** by default for SpringBoot web framework. We can define it as a dependency in the POM.XML file as shown in *SS 5.4.1* and is ready to be used.



```
49 <dependency>
50     <groupId>org.springframework.boot</groupId>
51     <artifactId>spring-boot-starter-test</artifactId>
52     <scope>test</scope>
53 </dependency>
```

*SS 5.4.1 Dependency for JUnit framework*

*SS 5.4.2* shows an example of unit test written for the application. While building or packaging into WAR file, all the test cases are run and checked than only the WAR file is build

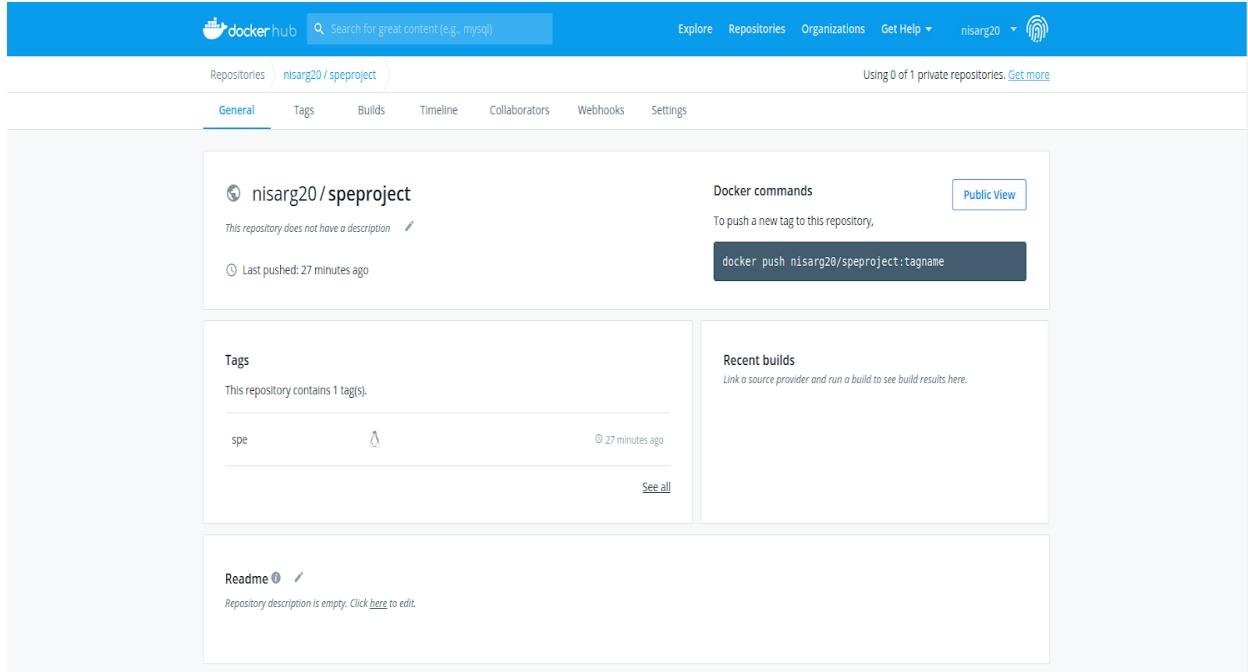


```
1 import com.spe.controller.UserController;
2 import org.junit.Before;
3 import org.junit.Test;
4 import org.junit.runner.RunWith;
5 import org.springframework.test.context.junit4.SpringRunner;
6 import org.springframework.test.web.servlet.MockMvc;
7 import org.springframework.test.web.servlet.setup.MockMvcBuilders;
8
9 import static org.springframework.test.web.servlet.request.MockMvcRequestBuilders.get;
10 import static org.springframework.test.web.servlet.result.MockMvcResultHandlers.print;
11 import static org.springframework.test.web.servlet.result.MockMvcResultMatchers.status;
12 import static org.springframework.test.web.servlet.result.MockMvcResultMatchers.view;
13
14 @RunWith(SpringRunner.class)
15 public class UserControllerTest {
16     private MockMvc mockMvc;
17
18     @Before
19     public void setUp() {
20         mockMvc = MockMvcBuilders.standaloneSetup(new UserController()).build();
21     }
22
23     @Test
24
25     public void testIndex() throws Exception{
26
27         this.mockMvc.perform(get( urlTemplate: "/"))
28
29             .andExpect(status().isOk())
30
31             .andExpect(view().name( expectedViewName: "index"))
32
33             .andDo(print());
34
35     }
36 }
```

*SS 5.4.2 Unit test example code*

## 5.5 Artifact

The artifact of our web application is the Docker Image created and pushed to DockerHub. *SS 5.5.1* shows the Docker Image on DockerHub.



*SS 5.5.1 Docker Image of the project on DockerHub*

The Docker Image is created by the deploy phase in the pipeline. The script is written in the **Dockerfile** located at the root directory of the project, running which the Docker Image is generated. *SS 5.5.2* shows the script in Dockerfile. It says that first install the openjdk onto the system, Expose port will bind the exposed port to the Docker host on a random port. Then the WAR file will be copied to the container and it will be run. [4]

```
1 ➤ FROM openjdk
2 EXPOSE 8085
3 ADD target/SPEProject.war SPEProject.war
4 ENTRYPOINT ["java" , "-jar" , "SPEProject.war"]
```

*SS 5.5.2 Dockerfile*

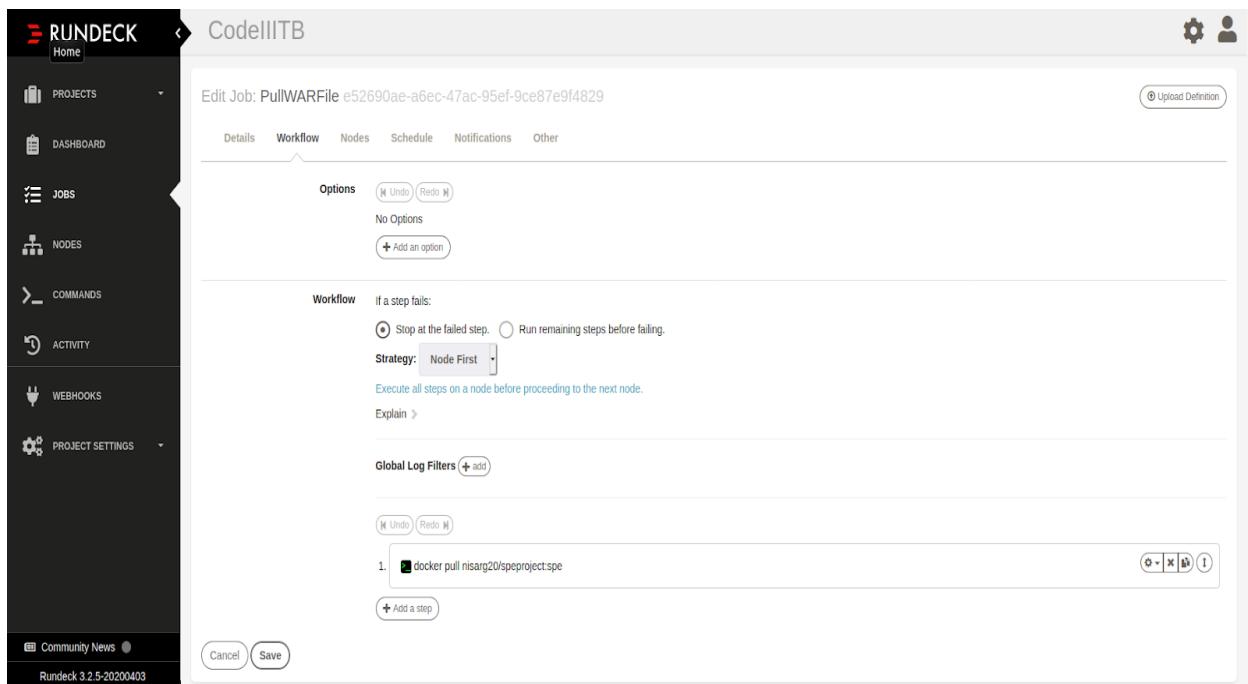
## 5.6 Deploy

Continuous deployment aims at minimizing the time elapsed between the development team, writing one new line of code, and this new code being used by live users, in production. We are using **Rundeck** as the **deploy and delivery tool**. Rundeck is an open source software that helps to automate routine operational procedures needed to carry out on thousands of nodes in data centers. [5]

Some of features of Rundeck are:

- Distributed command execution
- Workflow (including option passing, conditionals, error handling, and multiple workflow strategies)
- Pluggable execution system (SSH)
- Pluggable resource model (get details of your infrastructure from external systems)
- On-demand (Web GUI, API or CLI) or scheduled job execution

The Rundeck Job will be triggered from the pipeline in the delivery phase.



SS 5.6.1 Rundeck Job

SS 5.6.2 shows the successful run of the Rundeck Job. Here the Job is to pull the Docker Image from DockerHub.

The screenshot displays the Rundeck web interface. On the left, a dark sidebar contains navigation links: PROJECTS, DASHBOARD, JOBS, NODES, COMMANDS, ACTIVITY, WEBHOOKS, and PROJECT SETTINGS. Below these are links for Community News and a build status indicator. The main content area is titled "CodeIIITB". A job named "PullWARFile" is shown with a green checkmark icon and the status "Succeeded" followed by a timestamp "00:00:05 at 12:04 pm". A message "you" is displayed next to it. To the right is a "#173" link and a "Run Again" button. Below this, a "Log Output" section shows the execution details. It lists "100% 1/1 COMPLETE" with 0 FAILED, 0 INCOMPLETE, and 0 NOT STARTED steps. The "speproject" node has one step labeled "OK". The log output shows the command "spe: Pulling from nisarg20/speproject" with its digest and status. At the bottom, there are tabs for Stats, Activity, and a summary showing 7 EXECUTIONS, 71% SUCCESS RATE, and 12s AVG DURATION. The footer includes copyright information, unsupported software notice, and links for Licenses and Help.

SS 5.6.2 Successful Run of Rundeck Job

## 5.7 Monitoring

Monitoring is a backbone of the end-to-end delivery pipeline, and open source monitoring tools are like toppings on an ice cream scoop. It is desirable to monitor at almost every stage in order to have transparency about all the processes.

There are many tools available for monitoring like ELK stack, Zenoss and Nagios. We have used the ELK stack as a log analysis and monitoring tool. ELK is capable of collecting log data from heterogeneous sources and can scale as the data grows. It provides a wide variety of tools to analyse, understand and visualize the collected data with just a few clicks. [6]

**Elasticsearch** : A text search and analytics engine that stores all of the logs. It is capable of scaling to hundreds of servers ingesting petabytes of both structured and unstructured data.

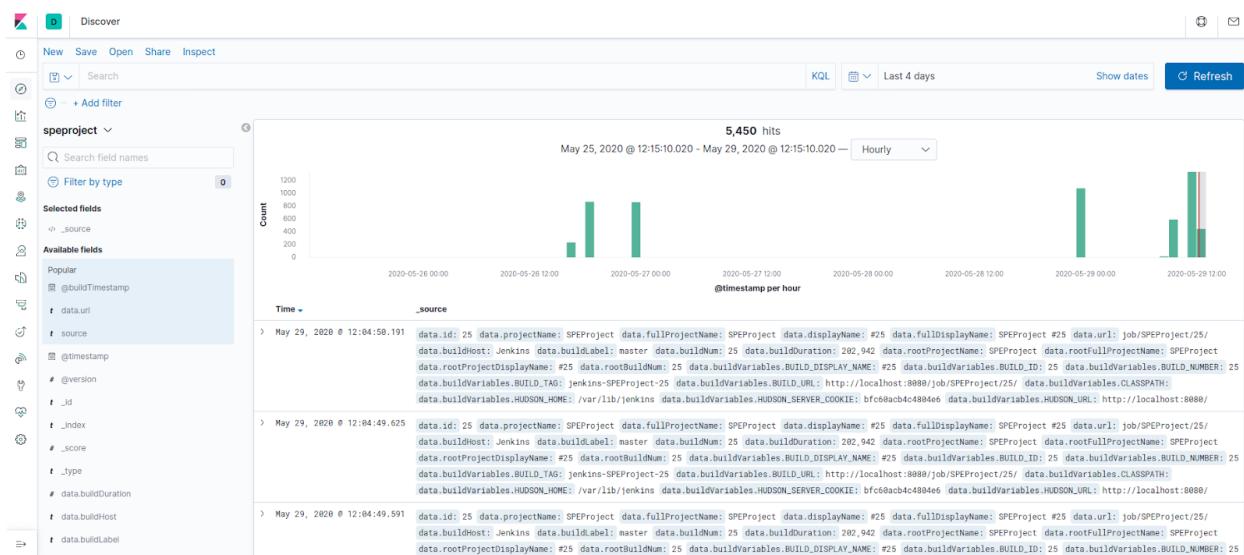
**Logstash** : The server component of that processes incoming logs

**Kibana** : Web interface for searching and visualizing logs collected over the ELK stack

The screenshot shows the Jenkins configuration interface for a Logstash job. The top navigation bar has 'Jenkins' and 'configuration'. Below it, there's a 'Logstash' section with a checked checkbox for 'Enable sending logs to an Indexer'. Under 'Indexer Type', 'Elastic Search' is selected with 'URI' set to 'http://localhost:9200/speproject/Nisarg'. The 'Password' field is marked as 'Concealed'. Other fields include 'User name' (empty), 'Mime Type' ('application/json'), and an 'Advanced...' button. Below this, there are sections for 'Enable Globally' (unchecked), 'Use millisecond time stamps' (checked), and 'Lockable Resources Manager'. Under 'Lockable Resources Manager', there's a 'Lockable Resources' section with an 'Add Lockable Resource' button. At the bottom, there are 'Save' and 'Apply' buttons.

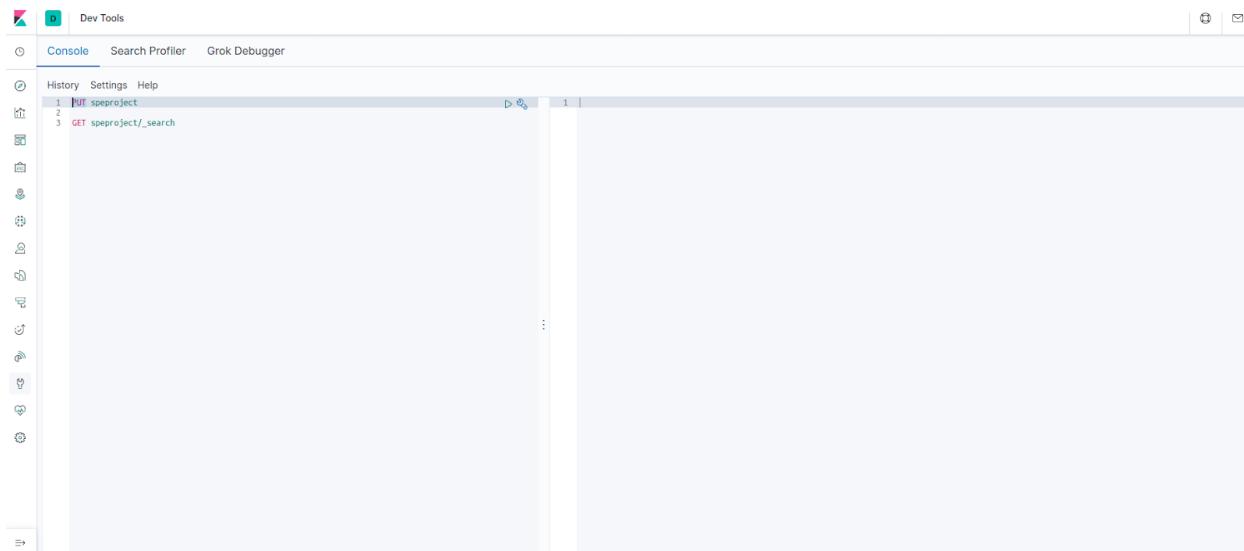
SS 5.7.1 Logstash configuration in Jenkins

SS 5.7.1 shows the configuration of Logstash to send the log data generated by Jenkins pipeline



SS 5.7.2 Timestamp view of logs

SS 5.7.3 shows how to create an Index. An index is a kind of Database



SS 5.7.3 Creating an Index

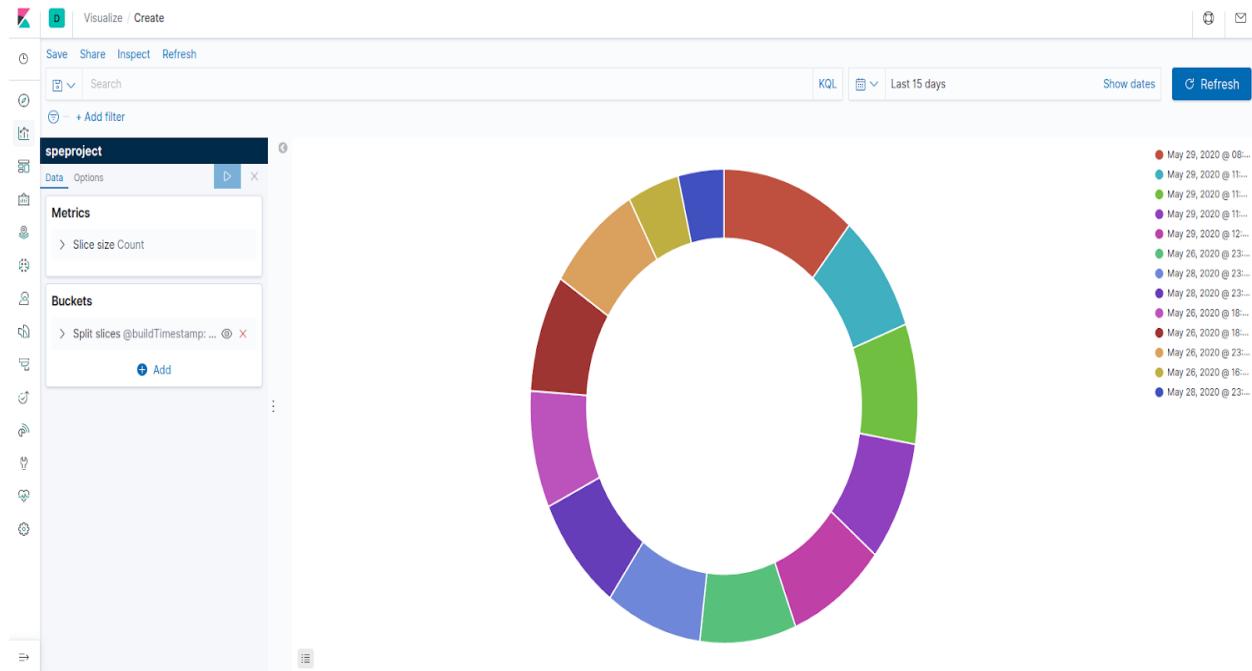
The screenshot shows the Elasticsearch Management interface with the 'Create index pattern' page open. On the left, there's a sidebar with 'Elasticsearch' and 'Kibana' sections. The main area is titled 'Create index pattern' with a sub-section 'Step 1 of 2: Define index pattern'. A text input field contains 'speroproject\*' with a note below it: 'You can use a \* as a wildcard in your index pattern. You can't use spaces or the characters \, /, ?, \*, <, >, |'. A success message says 'Your index pattern matches 1 index.' Below the input is a dropdown for 'Rows per page: 10'. At the bottom right is a 'Next step' button.

### SS 5.7.4 Creating an index pattern

This screenshot continues from the previous one, showing 'Step 2 of 2: Configure settings'. It shows the index pattern 'speroproject\*' has been defined. A 'Time Filter field name' dropdown is set to '@buildTimestamp'. A note explains: 'The Time Filter will use this field to filter your data by time. You can choose not to have a time field, but you will not be able to narrow down your data by a time range.' Below this is a 'Show advanced options' link. At the bottom right are 'Back' and 'Create index pattern' buttons.

### SS 5.7.5 Index configuration setting

SS 5.7.6 shows the Pie Chart of the build timestamp of the Pipeline.



SS 5.7.6 Visualization

## 6. THE CI/CD PIPELINE USING JENKINS

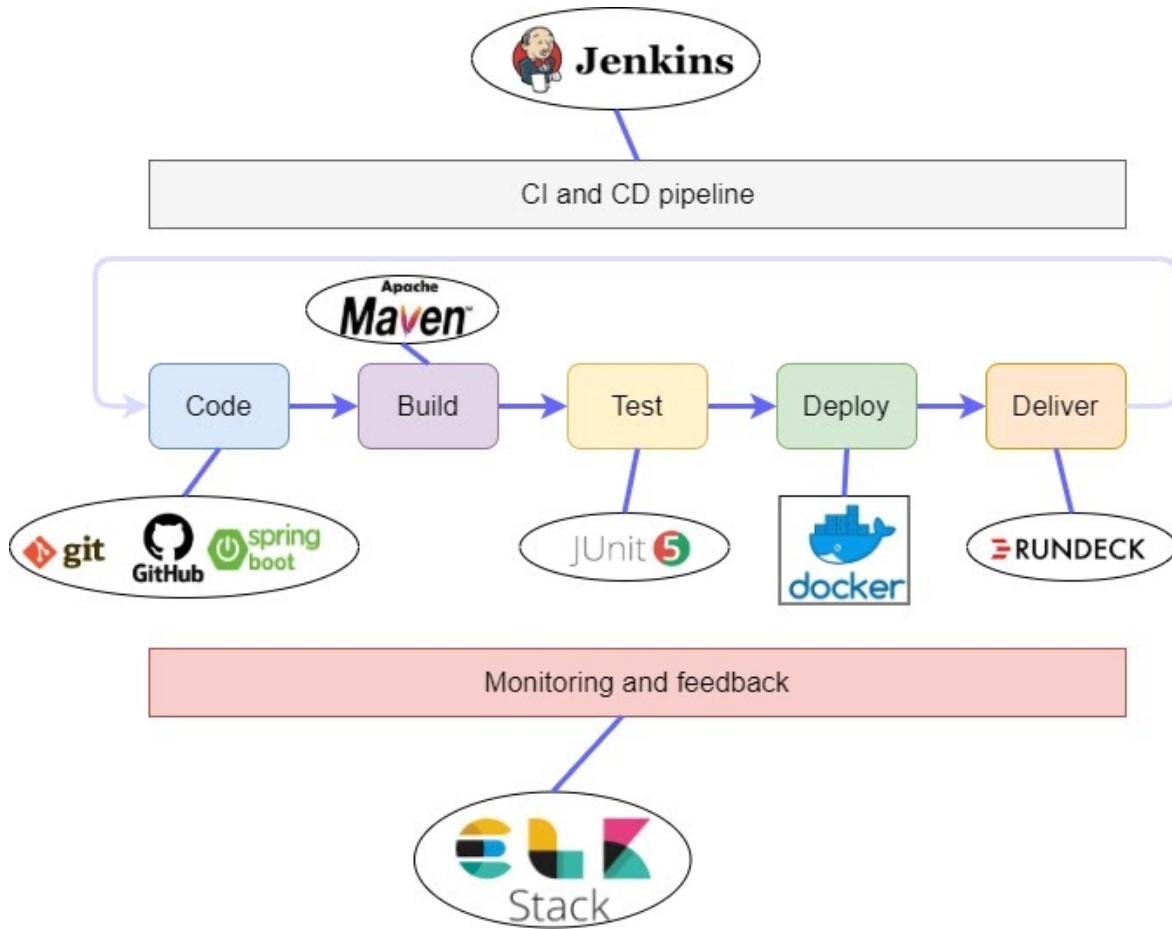


Figure - 6.1 The CI/CD pipeline of the project

CI is a software engineering practice where each check-in made by a developer is verified by either of the following:

**Pull mechanism** : Executing an automated build at a scheduled time

**Push mechanism** : Executing an automated build when changes are saved in the repository

This step is followed by executing a unit test against the latest changes available in the source code repository. CI makes things simple and helps us identify bugs or

errors in the code at a very early stage of development, when it is relatively easy to fix them.

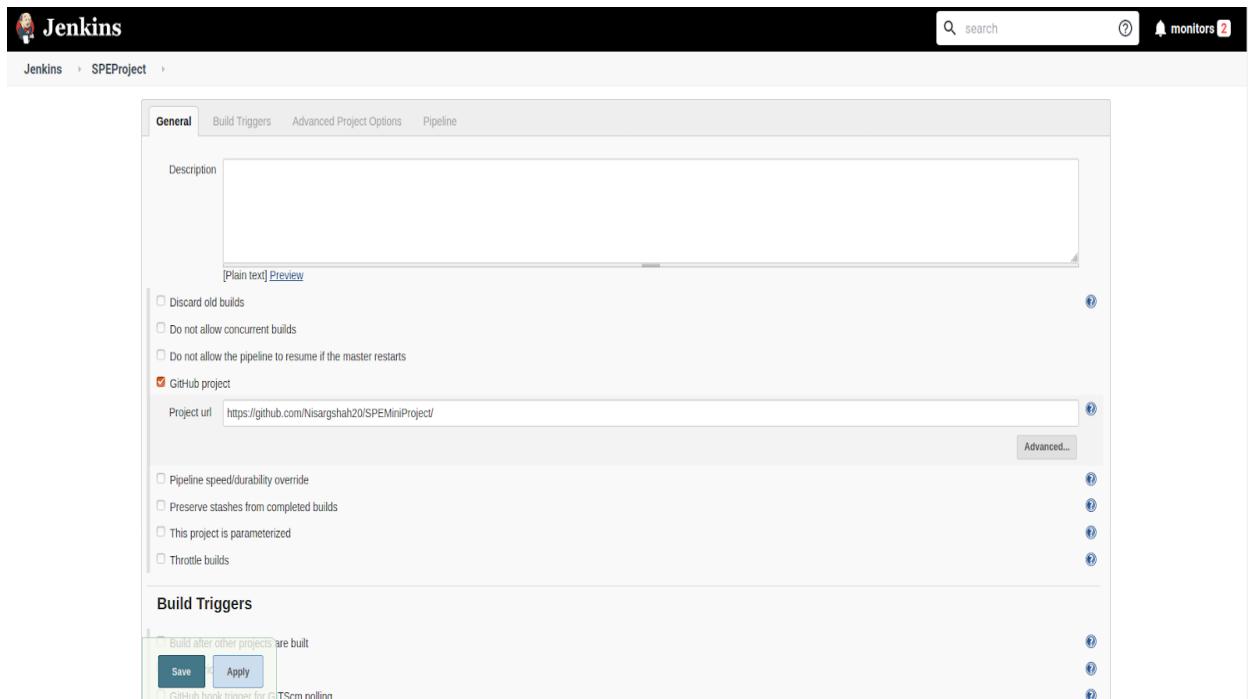
Jenkins is an open source and one the most popular CI tools available in the market. It helps in automating the repetitive task of CI. Jenkins makes the process effective and transparent. It is written in Java. It is easy to install and easy to configure. It provides extensibility with over 400 plugins for different integrations, such as the following:

- Source code management
- Build triggers
- Build reports
- Artifact uploaders
- External site/tool integrations
- UI plugins
- Authentication and user management
- Cluster management and distributed build

## 6.1 Creating Jenkins Pipeline

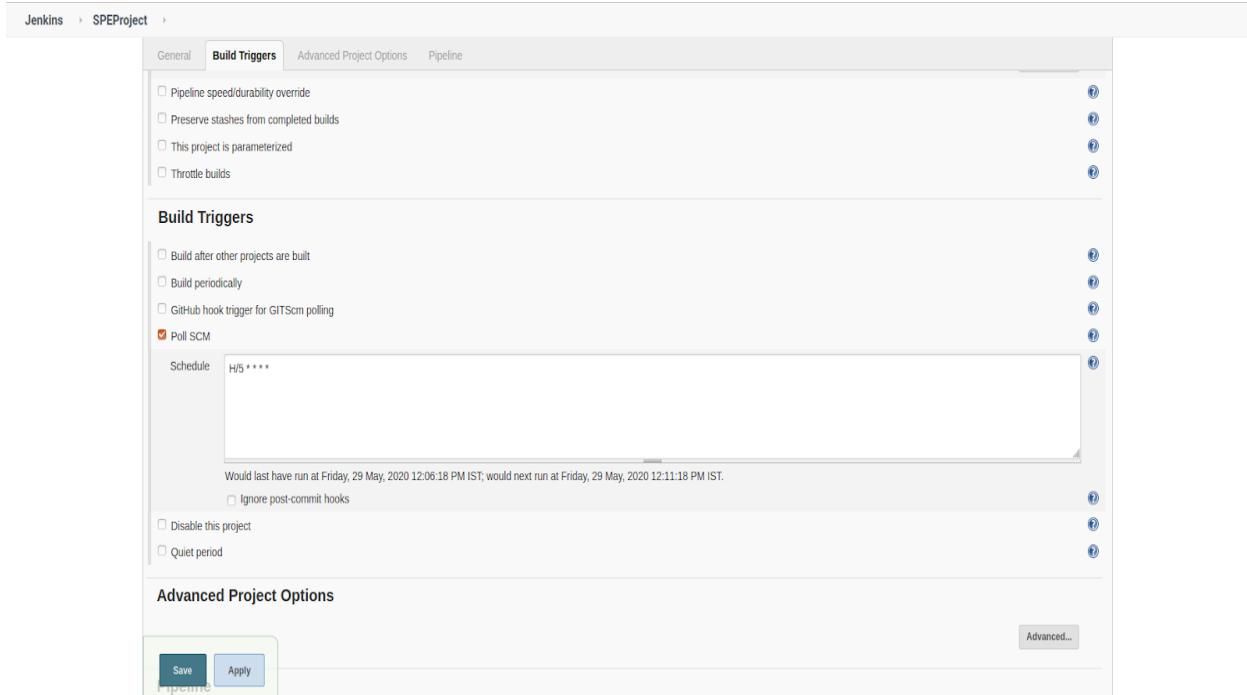
Before creating a pipeline project in Jenkins, we need to add necessary plugins needed for the project for example: Logstash, Git, Rundeck etc. Here we discuss how to setup the pipeline. Actually here we will just configure the pipeline. [3]

We create a new pipeline project by clicking on the new item on the left side of the dashboard in Jenkins and configure it as shown in below *SS 6.1.1, SS 6.1.2 and SS 6.1.3.*

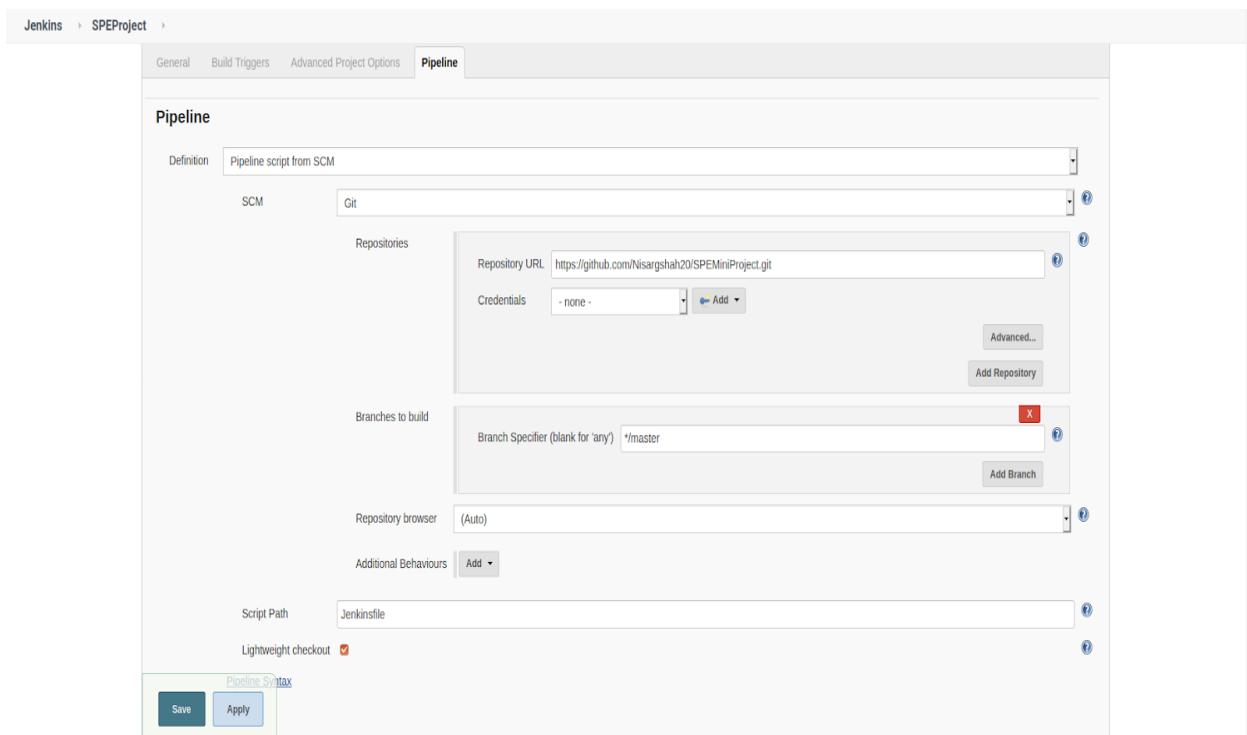


*SS 6.1.1 Providing github url for automatic checkout*

Here in SS 6.1.2, “H/5\*\*\*\*” means, scheduled the Jenkins to check the new pushes in the GitHub repository every 5 minutes. Hence, Jenkins will probe the changes in the code, every 5 minutes and if new commits are found then it will start the pipeline.



*SS 6.1.2 Configuring the pipeline build trigger*



### *SS 6.1.3 Providing location of pipeline script*

The actual pipeline phases will be written as script in the Jenkinsfile located at the root directory of the project. The *SS 6.1.4* shows the pipeline script written in the Jenkinsfile. We can decide what pipeline steps we want and code them. Here the steps are : Clean ->Compile(Package) -> Test -> Create and Deploy Docker Image to docker hub -> Execute the Rundeck Job. *Table 6.2.1* describes the function of each phases

```

1  pipeline {
2      environment {
3          registry = "nisarg20/speproject"
4          registryCredential = 'dockerhub'
5      }
6      agent any
7
8      stages {
9          stage('Clean') {
10             steps{
11                 logstash{
12                     sh 'mvn clean'
13                     echo "clean"
14                 }
15             }
16         }
17
18         stage('Compile') {
19             steps{
20                 logstash{
21                     sh 'mvn package'
22                     echo "compile"
23                 }
24             }
25         }
26
27         stage('Test') {
28             steps{
29                 logstash{
30                     sh 'mvn test'
31                     echo "test"
32                 }
33             }
34         }
35
36         stage('Deploy Image to Docker Hub') {
37             steps{
38                 logstash{
39                     script{
40                         dockerImage = docker.build registry + ":spe"
41                         docker.withRegistry( '', registryCredential){
42                             dockerImage.push()

```

```

43
44     }
45   }
46 }
47 }
48 stage('Execute Rundeck job') {
49   steps {
50     logstash{
51       script {
52         step([$class: "RundeckNotifier",
53           includeRundeckLogs: true,
54           jobId: "e52690ae-a6ec-47ac-95ef-9ce87e9f4829",
55           rundeckInstance: "Rundeck",
56           shouldFailTheBuild: true,
57           shouldWaitForRundeckJob: true,
58           tailLog: true])
59       }
60     }
61   }
62 }
63 }
64 }
```

*SS 6.1.4 Pipeline script*

Stage	Declarative: Checkout SCM	Clean	Compile	Test	Deploy Image to Docker Hub	Execute Rundeck job
#25	36s	23s	3min 28s	49s	1min 31s	1min 31s
#24	7s	12s	59s	31s	1min 26s	10s
#23	11s	16s	1min 52s	43s	1min 24s	11s
#22	12s	22s	1min 31s	44s	1min 20s	10s
#21	20s	15s	1min 5s	29s	1min 37s	43s
#20	14s	27s	2min 41s	1min 10s	1min 45s	13min 51s failed

*SS 6.1.5 Pipeline Stage View*

## 6.2 The Pipeline Phases

Phase	Task
Checkout SCM	The first phase is to pull the project from the GitHub repository and stores it in the Jenkins directory so than it can be used in the next phases
Clean	If there is previously compiled code in the directory and some classes are renamed in the latest build then this could cause problems. The previous compiled version will remain in target/classes until we clean the previous builds. Thus running clean makes sure the files that have been removed are discarded
Compile and Package	Package will compile the code and also package it i.e. if the project's POM.XML says that the project is a jar, it will create a jar or if the project is a war , it will create a war and put it somewhere in the target directory .
Test	Runs all the unit-test cases written in the test directory. Here all the test are written in JUnit(a java unit-test framework)
Deploy Docker Image to DockerHub	In this phase a new Docker Image is built according to the details of the provided in the Dockerfile and then pushes to the specified DockerHub registry
Execute Rundeck Job	This phase triggers the Rundeck to run the Job specified as the Job Id. Here the Job is to pull the Docker Image from DockerHub and run on the local node

Table 6.2 Pipeline Phases

## 6.3 Rundeck Integration in Jenkins

This can be done by going to Manage Jenkins->Configuration System and giving the instance name the same as Rundeck Instance in Jenkinsfile. The URL would be the URL of the Rundeck running server and provide the credentials of the admin account. It is shown in *SS 6.3.1*

The screenshot shows the Jenkins configuration interface under the 'Configuration System' section. On the left, there's a sidebar with links like 'New Item', 'People', 'Build History', 'Manage Jenkins', 'Lockable Resources', 'Credentials', and 'New View'. The main area has tabs for 'Build Queue' (which shows 'No builds in the queue.') and 'Build Executor Status' (which shows '1 Idle' and '2 Idle'). The right side is titled 'Rundeck' and contains two sections: 'Job cache' and 'Instances'. Under 'Job cache', there's a checkbox for 'Enable Rundeck job cache'. Under 'Instances', there's a table with one row for 'Rundeck'. The table columns are 'Name' (set to 'Rundeck'), 'URL' (set to 'http://localhost:4440'), 'Login' (set to 'admin'), 'Password' (set to 'Concealed' with a 'Change Password' button), 'Auth Token' (empty), and 'API Version' (empty). At the bottom of the page are 'Save' and 'Apply' buttons, and the URL 'localhost:8080'.

*SS 6.3.1 Rundeck Integration*

## 6.4 Logstash Integration with Jenkins

Here the only thing to add is URI which would be the URI of the Logstash running server like: **http://serveraddress:9200/index\_name/type**, where index\_name can be created using the dev tools option in Kibana using the command : **PUT index\_name** on the console. The index\_name must be in lowercase

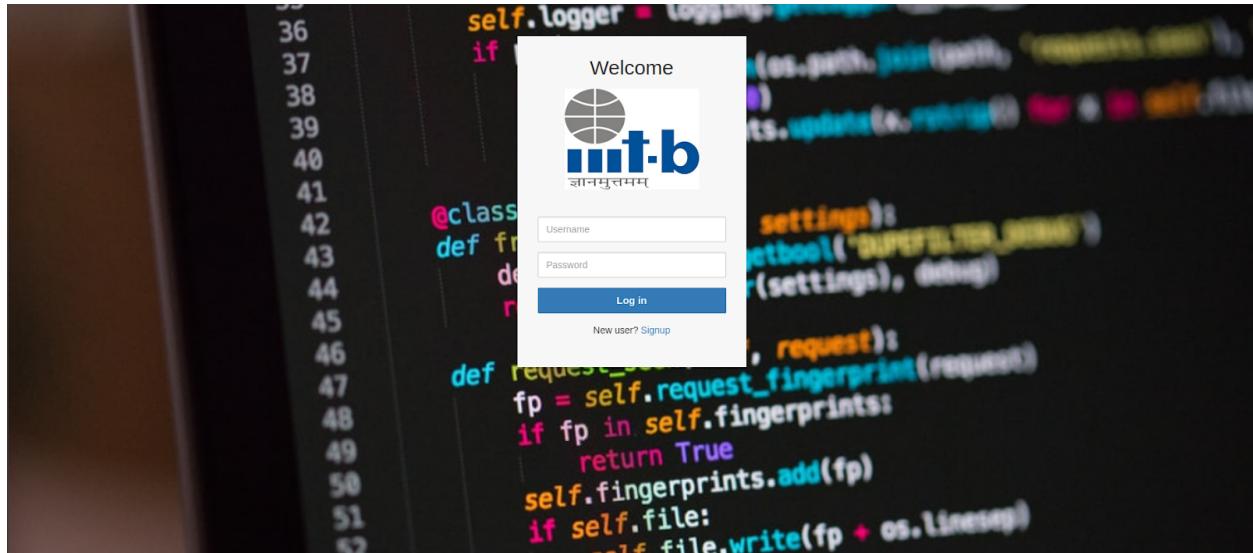
The screenshot shows the Jenkins configuration interface. The top navigation bar has 'Jenkins' and 'configuration' selected. A sub-menu 'Add' is open, showing 'Logstash'. The main configuration area for 'Logstash' is displayed, with the 'Enable sending logs to an Indexer' checkbox checked. Under 'Indexer Type', 'Elastic Search' is selected, with 'URI' set to 'http://localhost:9200/speproject/Nisarg', 'User name' empty, 'Password' concealed, and 'Mime Type' set to 'application/json'. There is an 'Advanced...' button. Below this, under 'Lockable Resources Manager', there is a 'Lockable Resources' section with an 'Add Lockable Resource' button. Under 'GitHub', there is a 'GitHub Servers' section with an 'Add GitHub Server' dropdown menu. At the bottom are 'Save' and 'Apply' buttons.

### SS 6.4.1 Logstash Integration

## 7. RESULT AND DISCUSSION

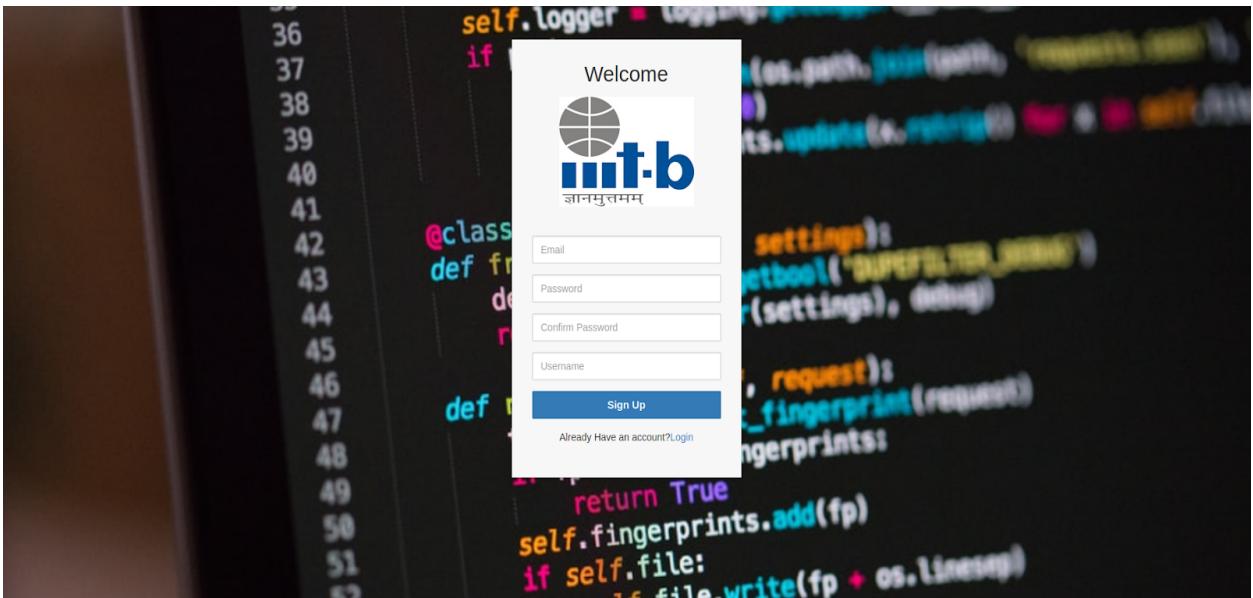
*Our Application Modules:*

- **Login:** It is the forest page of the application. Users log in to practice questions here.



SS 7.1 Login Page

- **Register:** If a user is first time visitor than he/she can register to system by providing the details and can login to use the application



SS 7.2 Registration Page

- **Home/Problems** : This is the home screen where the problems registered onto the system, for practice, are displayed. Users can choose any question and click on it to practice it. Clicking it will take the user to the IDE screen.

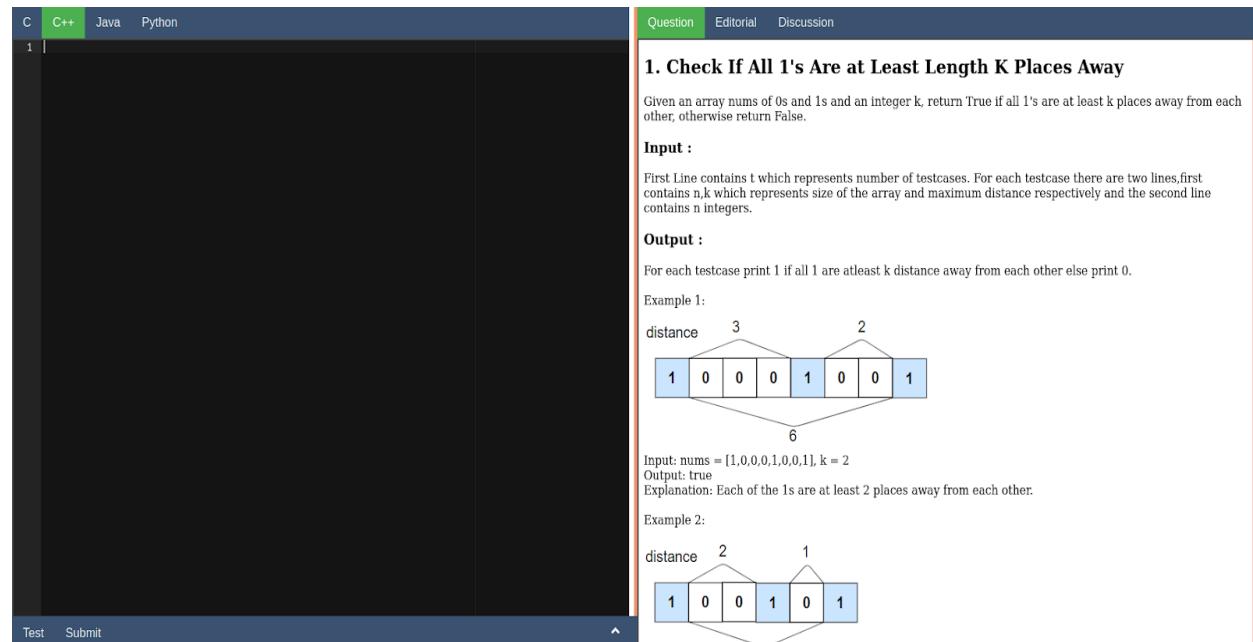


The screenshot shows a navigation bar at the top with links: Problems (highlighted in green), Upcoming Contests, Leaderboard, Discussion, Create Problem, and Host Contest. On the right, it shows the user's name: nisarg20. Below the navigation bar is a table with the following data:

SNo	Problem name	Difficulty	Topic	Accuracy
1	Check If All 1's Are at Least Length K Places Away	medium	array	60.0%
2	Remove Palindromic Subsequences	easy	string	54.54545454545454%

### SS 7.3 Problem List

- **Solve Problem** : This is the online IDE provide for writing code, running it and checking it whether users solution is correct or not



The screenshot shows the IDE interface on the left and the problem details on the right.

**IDE Interface:**

- Language tabs: C, C++, Java, Python (C++ is selected).
- Code editor area with the number "1" and a cursor.
- Buttons at the bottom: Test and Submit.

**Problem Details (Right Side):**

- Question Tab:** Contains the problem title: "1. Check If All 1's Are at Least Length K Places Away".
- Editorial Tab:** Contains the problem statement: "Given an array nums of 0s and 1s and an integer k, return True if all 1's are at least k places away from each other, otherwise return False."
- Discussion Tab:** Not visible in the screenshot.
- Input:** Describes the input format: "First Line contains t which represents number of testcases. For each testcase there are two lines, first contains n,k which represents size of the array and maximum distance respectively and the second line contains n integers."
- Output:** Describes the output format: "For each testcase print 1 if all 1 are atleast k distance away from each other else print 0."
- Example 1:** Shows an array [1, 0, 0, 0, 1, 0, 0, 1] with a maximum distance of 3. A diagram shows the array with nodes connected by lines representing distances between 1s. The distance between the first and third 1s is 3, and between the third and fourth 1s is 2.
- Example 2:** Shows an array [1, 0, 0, 1, 0, 1] with a maximum distance of 2. A diagram shows the array with nodes connected by lines representing distances between 1s. The distance between the first and third 1s is 2, and between the third and fourth 1s is 1.
- Explanation:** For Example 1, the explanation is: "Each of the 1s are at least 2 places away from each other."

### SS 7.4 The IDE and Problem Page

Here is an example of code written in the IDE

C C++ Java Python
Question Editorial Discussion

```

1 #include <iostream>
2 #include <bits/stdc++.h>
3 using namespace std;
4
5 bool kLengthApart(vector<int>& nums, int k)
6 {
7
8     int start = -1;
9     for(int i=0;i<nums.size();i++)
10    {
11        if(nums[i]==1)
12        {
13            if(start<=i && i-start<=k) return 0;
14            start = i;
15        }
16    }
17    return 1;
18 }
19 int main()
20 {
21     int t;
22     cin>>t;
23     while(t--)
24     {
25         int n,k;
26         cin>>n>>k;
27         int t,j;
28         vector<int>nums;
29         for(i=0;i<n;i++)
30         {
31             cin>>j;nums.push_back(j);
32         }
33         cout<<kLengthApart(nums,k)<<endl;
34     }
35 }
36
37

```

Test
Submit

**1. Check If All 1's Are at Least Length K Places Away**

Given an array nums of 0s and 1s and an integer k, return True if all 1's are at least k places away from each other, otherwise return False.

**Input :**

First Line contains t which represents number of testcases. For each testcase there are two lines,first contains n,k which represents size of the array and maximum distance respectively and the second line contains n integers.

**Output :**

For each testcase print 1 if all 1 are atleast k distance away from each other else print 0.

Example 1:

Input: nums = [1,0,0,0,1,0,0,1], k = 2  
Output: true  
Explanation: Each of the 1s are at least 2 places away from each other.

Example 2:

Input: nums = [1,0,0,1,0,1], k = 2  
Output: true  
Explanation: Each of the 1s are at least 2 places away from each other.

### SS 7.5 Code in IDE

Clicking the test button will compile the entered code and accordingly display output in the output tab

C C++ Java Python
Question Editorial Discussion

```

1 #include <iostream>
2 #include <bits/stdc++.h>
3 using namespace std;
4
5 bool kLengthApart(vector<int>& nums, int k)
6 {
7
8     int start = -1;
9     for(int i=0;i<nums.size();i++)
10    {
11        if(nums[i]==1)
12        {
13            if(start<=i && i-start<=k) return 0;
14            start = i;
15        }
16    }
17    return 1;
18 }
19 int main()
20 {
21     int t;
22     cin>>t;
23     while(t--)
24     {
25         int n,k;
26         cin>>n>>k;
27         int t,j;
28         vector<int>nums;
29         for(i=0;i<n;i++)
30         {
31             cin>>j;nums.push_back(j);
32         }
33         cout<<kLengthApart(nums,k)<<endl;
34     }
35 }
36
37

```

Test
Submit

**1. Check If All 1's Are at Least Length K Places Away**

Given an array nums of 0s and 1s and an integer k, return True if all 1's are at least k places away from each other, otherwise return False.

**Input :**

First Line contains t which represents number of testcases. For each testcase there are two lines,first contains n,k which represents size of the array and maximum distance respectively and the second line contains n integers.

**Output :**

For each testcase print 1 if all 1 are atleast k distance away from each other else print 0.

Example 1:

Input: nums = [1,0,0,0,1,0,0,1], k = 2  
Output: true  
Explanation: Each of the 1s are at least 2 places away from each other.

Example 2:

Input: nums = [1,0,0,1,0,1], k = 2  
Output: true  
Explanation: Each of the 1s are at least 2 places away from each other.

### SS 7.6 Testing the Code

For a correct answer a status “Right answer” will be displayed as shown in here

C C++ Java Python

```

1 #include <iostream>
2 #include <bits/stdc++.h>
3 using namespace std;
4
5 bool kLengthApart(vector<int>& nums, int k)
6 {
7     int start = -1;
8     for(int i=0;i<nums.size();i++)
9     {
10        if(nums[i]==1)
11        {
12            if(start<=1 && i-start<=k) return 0;
13            start = i;
14        }
15    }
16    return 1;
17 }
18
19 int main()
20 {
21     int t;
22     cin>>t;
23     while(t--)
24     {
25         int n,k;
26         cin>>n>>k;
27         int t,j;
28         vector<int> nums;
29         for(i=0;i<n;i++)
30             ...
31     }
32 }
```

Test Submit Right Answer

Input
Output

```
1
5 0
1 1 1 1
```

```
1
```

**1. Check If All 1's Are at Least Length K Places Away**

Given an array nums of 0s and 1s and an integer k, return True if all 1's are at least k places away from each other, otherwise return False.

**Input :**

First Line contains t which represents number of testcases. For each testcase there are two lines, first contains n,k which represents size of the array and maximum distance respectively and the second line contains n integers.

**Output :**

For each testcase print 1 if all 1 are atleast k distance away from each other else print 0.

Example 1:

Input: nums = [1,0,0,0,1,0,0,1], k = 2  
Output: true  
Explanation: Each of the 1s are at least 2 places away from each other.

Example 2:

Input: nums = [1,0,0,1,0,1], k = 2  
Output: true  
Explanation: Each of the 1s are at least 2 places away from each other.

SS 7.7 Correct Output

If there is some compilation error then it will show as following

C C++ Java Python

```

1 #include <iostream>
2 #include <bits/stdc++.h>
3 using namespace std;
4
5 bool kLengthApart(vector<int>& nums, int k)
6 {
7     int start = -1;
8     for(int i=0;i<nums.size();i++)
9     {
10        if(nums[i]==1)
11        {
12            if(start<=1 && i-start<=k) return 0;
13            start = i;
14        }
15    }
16    return 1;
17 }
18
19 int main()
20 {
21     int t;
22     cin>>t;
23     while(t--)
24     {
25         int n,k;
26         cin>>n>>k;
27         int t,j;
28         vector<int> nums;
29         for(i=0;i<n;i++)
30             ...
31     }
32 }
```

Test Submit ? Compilation Error

```
src/main/resources/nsisarg20/1/userprogram.cpp: In function 'bool kLengthApart(std::vector<int>, int)':
src/main/resources/nsisarg20/1/userprogram.cpp:9:9: error: expected ';' or ';' before 'for'
    for(int i=0;i<nums.size();i++)
    ^
src/main/resources/nsisarg20/1/userprogram.cpp:9:21: error: 'i' was not declared in this scope
    for(int i=0;i<nums.size();i++)
    ^
```

Input
Output

```
1
5 0
1 1 1 1
```

```
1
```

**1. Check If All 1's Are at Least Length K Places Away**

Given an array nums of 0s and 1s and an integer k, return True if all 1's are at least k places away from each other, otherwise return False.

**Input :**

First Line contains t which represents number of testcases. For each testcase there are two lines, first contains n,k which represents size of the array and maximum distance respectively and the second line contains n integers.

**Output :**

For each testcase print 1 if all 1 are atleast k distance away from each other else print 0.

Example 1:

Input: nums = [1,0,0,0,1,0,0,1], k = 2  
Output: true  
Explanation: Each of the 1s are at least 2 places away from each other.

Example 2:

Input: nums = [1,0,0,1,0,1], k = 2  
Output: true  
Explanation: Each of the 1s are at least 2 places away from each other.

SS 7.8 Compilation Error

It shows the screen when a user submits wrong solution

C C++ Java Python

```

1 #include <iostream>
2 #include <bits/stdc++.h>
3 using namespace std;
4
5 bool kLengthApart(vector<int>& nums, int k)
6 {
7
8     int start = -1;
9     for(int i=0;i<nums.size();i++)
10    {
11        if(nums[i]==1)
12        {
13            if((start+1 == i) && i-start <= k) return 0;
14            start = i;
15        }
16    }
17    return 1;
18 }
19 int main()
20 {
21     int t;
22     cin>>t;
23     while(t--)
24     {
25         int n,k;
26         cin>>n>>k;
27         int i,j;
28         vector<int>nums;
29         for(i=0;i<n;i++)
30     
```

Test Submit ❌ Wrong Answer

Input	Output
1 5 0 1 1 1 1 1	1

Question Editorial Discussion

### 1. Check If All 1's Are at Least Length K Places Away

Given an array nums of 0s and 1s and an integer k, return True if all 1's are at least k places away from each other, otherwise return False.

**Input :**

First Line contains t which represents number of testcases. For each testcase there are two lines, first contains n,k which represents size of the array and maximum distance respectively and the second line contains n integers.

**Output :**

For each testcase print 1 if all 1 are atleast k distance away from each other else print 0.

**Example 1:**

Input: nums = [1,0,0,0,1,0,0,1], k = 2  
Output: true  
Explanation: Each of the 1s are at least 2 places away from each other.

**Example 2:**

### SS 7.9 Wrong Answer

If user wants to checkout solution than it can be viewed by clicking on editorial tab

C C++ Java Python Zoom-In Zoom-Out

```

1 #include <iostream>
2 #include <bits/stdc++.h>
3 using namespace std;
4
5 int main()
6 {
7     int t;
8     cin>>t;
9     while(t--)
10    {
11    }
12 }
13
14

```

Test Submit

Question Editorial Discussion

```

#include <iostream>
#include <bits/stdc++.h>
using namespace std;

bool kLengthApart(vector<int>& nums, int k)
{
    int start = -1;
    for(int i=0;i<nums.size();i++)
    {
        if(nums[i]==1)
        {
            if(start+1 == i) return 0;
            start = i;
        }
    }
    return 1;
}

int main()
{
    int t;
    cin>>t;
    while(t--)
    {
        int n,k;
        cin>>n>>k;
        int i,j;
        vector<int>nums;
        for(i=0;i<n;i++)
        {
            cin>>j;
            nums.push_back(j);
        }
        cout<<kLengthApart(nums,k)<<endl;
    }
}

```

### SS 7.10 Editorial Page

- **Upcoming Contests:** All the upcoming coding contests are displayed here

The screenshot shows a navigation bar with links: Problems, Upcoming Contests (highlighted in green), Leaderboard, Discussion, Create Problem, and Host Contest. On the right, it says "nisarg20". Below the navigation bar is a table with four columns: Name, Start Time, Date, and Length. The data is as follows:

Name	Start Time	Date	Length
IITB Code	12:00	28/05/2020	2 hours
Snackdown	21:30	30/05/2020	2:30 hours
Hackathon	21:00	31/05/2020	3 hours

### *SS 7.11 Upcoming Contests*

- **Leaderboard :** The user submits their solutions and accordingly they get points which reflects them on the leaderboard screen showing where they stand among all users

The screenshot shows a navigation bar with links: Problems, Upcoming Contests, Leaderboard (highlighted in green), Discussion, Create Problem, and Host Contest. On the right, it says "nisarg20". Below the navigation bar is a table with six columns: SNo, UserName, Easy, Medium, Hard, and Points. The data is as follows:

SNo	UserName	Easy	Medium	Hard	Points
1	nisarg41	2	4	2	55
2	rachit41	2	5	3	45
3	nirmal29	1	3	1	22
4	rachit20	0	0	0	0
5	nisarg20	0	0	0	0

### *SS 7.12 Leaderboard Page*

- **Discussion :** If a user want to participate in some discussion or ask some doubt regarding problem, he/she can do it using discussion tab

The screenshot shows a discussion thread on a platform. At the top, there are navigation tabs: Problems, Upcoming Contests, Leaderboard, Discussion (which is highlighted in green), Create Problem, and Host Contest. On the right, the user 'nisarg20' is logged in.

**nisarg20**  
Posted on 2020-05-28 22:18:03.0  
Thanks

**rachit20**  
Posted on 2020-05-28 20:51:15.0  
Nice Solution @Nisarg20

**nisarg20**  
Posted on 2020-05-28 20:50:01.0  
Explanation of Remove Palindromic subsequence:  
This feels like almost a trick question. There are only 3 key ideas to realize to formulate the solution:  
1. For any random string consisting of only a's and b's, we can first remove all the a's (aaa...a is a palindrome) and then we can remove all the remaining b's (bbb...b). This means our answer cannot be > 2.  
2. Additionally, if our string is already a palindrome, we only need 1 operation, just remove the string itself.  
3. If the string is empty, then we do not need to reduce it at all (0 operations).

On the bottom left, there is a note: `Can we simulate check that for the empty string case and palindromic string and return 2 in all other cases.`

### SS 7.13 Discussion Page

- **Create Problems :** If user want to submit some good questions than he can do that with submit question tab

The screenshot shows the 'Create Problem' page. At the top, there are navigation tabs: Problems, Upcoming Contests, Leaderboard, Discussion, Create Problem (which is highlighted in green), and Host Contest. On the right, the user 'nisarg20' is logged in.

**Submit a problem**

**Problem Description**

**Sample Input**

**Sample Output**

**Submit**

### SS 7.14 Create Problem Page

- **Host a contest :** User can create a new contest here

The screenshot shows a web application interface for hosting a contest. At the top, there is a navigation bar with links: Problems, Upcoming Contests, Leaderboard, Discussion, Create Problem, and Host Contest. The 'Host Contest' link is highlighted in green. On the right side of the header, the user's name 'nisarg20' is displayed. Below the header, the page title 'Host a Contest' is centered. There are several input fields for contest configuration: 'Problem Name' (empty), 'Contest Length' (empty), 'Date' (dd / mm / yyyy format, empty), and 'Start Time' (empty). Below these fields are two buttons: 'Add Question' and 'Remove Question'. A table is present for managing contest questions, with columns: Time Limit, Question, Input, Output, and Points. The first row of the table has a file input field for the question file, which is currently empty ('No file selected.'), and a 'Browse...' button. The other three columns for this row are also empty. A 'Submit' button is located at the bottom left of the form.

Time Limit	Question	Input	Output	Points
<input type="file"/>	<input type="button" value="Browse..."/> No file selected.	<input type="button" value="Browse..."/> No file selected.	<input type="button" value="Browse..."/> No file selected.	<input type="text"/>

*SS 7.15 Host Contest Page*

## **8. SCOPE FOR FUTURE WORK**

Our current work includes an online IDE, editorials, discussion forum and leaderboard. We will be adding the Live Competition and Competition creation in future. About competitions there can be extensions to the traditional coding competitions like: One on One Codewars and Team Codewars

Currently our Judge is on a local system but must be made in such a way that it can scale load of thousands of users at least.

## **9. CONCLUSION**

Our web application provides a platform to practice, compete, discuss coding questions online with no extra installation or other resource. We used the Industry Standard process: Devops for developing this project so that it can easily scale, new features can be frequently released and delivered to the users. We showed the CI/CD pipeline for the project development. We used different tools like Jenkins, Rundeck, ELK stack and learned how a large scale project can be implemented with help of using several DevOps tools.

## 10. REFERENCES

1. SpringBoot Reference Documentation. [online] Available at:  
<https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/>
2. Getting Started with Amazon RDS - Amazon Relational Database Service. Available at :  
[https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP\\_GettingStarted.html](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_GettingStarted.html)
3. Jenkins User Documentation. [online] Available at :  
<https://www.jenkins.io/doc/>
4. Dockerfile reference | Docker Documentation. [online] Available at :  
<https://docs.docker.com/engine/reference/builder/>
5. Configurations | Rundeck Docs. [online] Available at :  
<https://docs.rundeck.com/docs/manual/09-configure.html>
6. The Complete Guide to the ELK Stack | Logz.io. [online] Available at :  
<https://logz.io/learn/complete-guide-elk-stack/#logstash>