System Requirements Specification Index

For

Git Tags With Branches

Version 1.0

IIHT Pvt. Ltd.

fullstack@iiht.com

TABLE OF CONTENTS

Git Tags With Branches

System Requirements Specification

1 Project Abstract

This document outlines the structure for a **Git Tags with Branches git assessment** designed to evaluate the candidate's proficiency in using Git commands, integrating these commands within a Java application, and managing the build process with Maven. The assessment involves executing specified Git commands, verifying their correctness through a Java application, and using Maven to build and test the application.

2 Assessment Objectives

The objective of this assessment is to test the candidate's ability to utilize git commands effectively with a project environment.

3 Assessment Tasks

1. Open the terminal in the parent folder. Ensure that the folder path matches the email ID you used for the assessment.
2. Create a file named file.txt and add some content to it i.e "dummy text".
3. Stage the file (file.txt) and commit it with the message "adding file.txt".
4. Create and checkout to the discount branch.
5. Create a file named discount.txt and add content to it i.e "added logic for giving discount".
6. Stage the file (discount.txt) and commit it with the message "Add discount coupons feature".
7. Checkout to the main branch.
8. Merge the discount branch into main.
9. Create a tag v1.1.0 with the message "Release v1.1.0: Discount Coupons Feature".
10. Checkout to the main branch again.
11. Create a new branch hotfix-payment-gateway.
12. Create a file named hotfix.txt and add content to it i.e "fixed payment gateway issue".
13. Stage the file (hotfix.txt) and commit it with the message "Fix payment gateway issue".
14. Create a tag v1.1.1 with the message "Hotfix v1.1.1: Payment Gateway Issue".
15. Checkout to the develop branch.
16. Merge the hotfix-payment-gateway branch into develop.
17. Checkout to the develop branch and create a tag v1.2.0-beta with the message "Pre-release v1.2.0-beta: Features under development".

4 EXECUTION STEPS TO FOLLOW

1. To open the command terminal, you need to go to the Application menu (Three horizontal lines at left top) -> Terminal -> New Terminal.
2. Once you perform all tasks, please open another terminal with the root address (path with project name).
3. To run your project use command:

```
mvn clean install exec:java -Dexec.mainClass="mainapp.MyApp" -DskipTests=true
```

4. To test your project, use the command

```
mvn test
```

5. If you want to exit(logout) and continue the coding later anytime (using Save & Exit option on Assessment Landing Page) then you need to use CTRL+Shift+B-command compulsorily on code IDE. This will push or save the updated contents in the internal git/repository. Else the code will not be available in the next login.
6. These are time bound assessments the timer would stop if you logout and while logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.
7. You need to use **CTRL+Shift+B** - command compulsorily on code IDE, before final submission as well. This will push or save the updated contents in the internal git/repository, and will be used to evaluate the code quality.