System Requirements Specification Index

For

Git Merging

Version 1.0

IIHT Pvt. Ltd.
fullstack@iiht.com

TABLE OF CONTENTS

Git Merging

System Requirements Specification

1 PROJECT ABSTRACT

This document outlines the structure for a **Git Merging 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. Initialize an empty Git repository.
3. Create a new file named index.html and add some initial content to it i.e, "add some code in index.html file".
4. Stage and commit the changes with the message "adding index.html file"
5. Create and checkout to the blog-feature branch.
6. Modify index.html to add content for the blog section i.e "update index.html file with a code for blog section".
7. Stage and commit the changes with the message "added blog feature".
8. Checkout to the main branch.
9. Merge the blog-feature branch into main using a fast-forward merge .
10. Checkout to the comment-feature branch.
11. Modify index.html to add content for the comment section i.e "update index.html file with a code for comment section".
12. Stage and commit the changes with the message "added comment section".
13. Checkout to the main branch.
14. Modify index.html to add content for the "About Us" section i.e "update index.html file with a code for about us section".
15. Stage and commit the changes with the message "added about us".
16. Checkout back to the main branch.
17. Merge the comment-feature branch into main.

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.