System Requirements Specification Index

For

Git Merging Branches

Version 1.0

IIHT Pvt. Ltd.
fullstack@iiht.com

TABLE OF CONTENTS

1 PROJECT ABSTRACT

This document outlines the structure for a **Merging 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. Initialize an empty git repository inside utils folder.
2. Utils folder contains Assessment.txt file already.
3. Edit the file and add your name in Assessment.txt file.
4. Add this file to git base and commit the changes with comment “my name”.
5. Create a new branch from master branch with name tmpbranch.
6. Checkout to tmpbranch.
7. Edit the Assessment.txt file of tmpbranch and add your contact number.
8. Commit the changes with comment “my number”.
9. Merge tmpbranch back into master branch.

4 EXECUTION STEPS TO FOLLOW

1. **All git commands must be executed only in utils folder.**
2. **To open the command terminal, you need to go to the Application menu (Three horizontal lines at left top) -> Terminal -> New Terminal and then change the path to utils folder.**
3. **Once you perform all tasks, please open another terminal with root address (path with project name).**
4. **To run your project use command:**

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

5. **To test your project, use the command**

   ```
   mvn test
   ```

6. **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.**
7. **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.**
8. **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.**