

---

# System Requirements Specification Index

For

**Git Rebase**

Version 1.0

**IIHT Pvt. Ltd.**

[fullstack@iiht.com](mailto:fullstack@iiht.com)

# TABLE OF CONTENTS

|   |                       |   |
|---|-----------------------|---|
| 1 | Project Abstract      | 3 |
| 2 | Assessment Objectives | 3 |
| 3 | Assessment Tasks      | 3 |
| 4 | Execution Steps       | 3 |

# Git Rebase

## System Requirements Specification

### 1 PROJECT ABSTRACT

This document outlines the structure for a **Git Rebase 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 file named file.txt and add the content "First commit on main".
4. Stage and commit the changes with the message "Initial commit on main".
5. Create and checkout to the feature-branch branch.
6. Create a new file feature.txt and add the content "Feature work 1".
7. Stage and commit the file with the message "Add feature work 1".
8. Modify feature.txt by appending the content "Feature work 2".
9. Stage and commit the changes with the message "Add feature work 2".
10. Checkout to the main branch.
11. Modify file.txt by adding the content "Changes on main".
12. Stage and commit the changes with the message "Changes made on main".
13. Checkout to the main branch.
14. Stage and commit all files with the message "adding all files".
15. Rebase feature-branch onto main to integrate the changes from feature-branch into the main branch.

### 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.