



National University
of computer and emerging sciences

Assignment #3

Course:

Software Quality Engineering

Topic:

LAYP
(UNIT Testing)

Submitted by:

Roll No.

22F-3714

22F-3722

Name

Faizan Tariq

Muhammad Abdullah

Submitted to:

Mr. Jawad Khalid

Contents

Introduction	3
Project Overview	3
Objective	3
Features	3
Setup and Installation	3
Installation Steps	3
Testing Strategy	4
Test Designs	4
Running Tests	8
Technology Stack	8

Introduction

LAYP-Real Editor is a lightweight, scalable, and feature-rich text editor framework built to simplify document editing for end-users. It provides advanced text formatting options, file management capabilities, and an intuitive user interface suitable for developers and non-technical users alike.

Project Overview

LAYP-RealEditor was developed as part of the LAYP framework initiative. It is designed for extensibility and adheres to best practices in software engineering. The primary focus is to provide a seamless experience for document creation and editing, with support for integrations and modular enhancements.

Objective

The primary goal of LAYP-RealEditor is to:

- Enhance productivity through simplified editing workflows.
 - Provide a flexible architecture for future extensions.
 - Support multiple document formats (e.g., .txt, .md, .docx).
-

Features

- **Rich Text Editing:** Apply bold, italic, underline, and other styles effortlessly.
 - **File Management:** Save, open, and export files in multiple formats.
 - **Customizable Themes:** Choose between light, dark, or custom themes.
 - **Keyboard Shortcuts:** Enhanced accessibility with configurable shortcuts.
 - **Version Control:** Built-in version tracking for editing history.
 - **Plugin Support:** Extend functionality with custom plugins.
-

Setup and Installation

Prerequisites

- Node.js (version 16 or later)
- Git
- A modern browser (Chrome, Firefox, Edge)

Installation Steps

1. Clone the repository:

➤ `git clone https://github.com/whis-19/LAYP-RealEditor.git`
`cd LAYP-RealEditor`

Usage Instructions

1. Create a Document

- Click File > New to create a new document.

2. Read/Open a Document

- Click File > Open to upload and view an existing document.

3. Update/Edit a Document

- Use the text editor to modify the document's content.
- Apply formatting as needed using the toolbar.

4. Delete a Document

- Remove unwanted documents directly from your file system or use File > Delete (if supported by your configuration).
-

Testing Strategy

The testing strategy follows a modular approach:

1. **Unit Testing:** Each function and component is tested in isolation using Jest.
2. **Integration Testing:** Interactions between major modules are tested using Playwright.
3. **End-to-End Testing:** Simulate real-world user interactions with CodeceptJS and Allure for reporting.

Test Designs

Test ID	Test Description	Category	Inputs	Expected Result	Actual Result
1.	Return non-empty list when files exist	ECP	Files exist in DB	Returns non-empty list	Pass
2.	Create file with special characters in filename	ECP	Filename: File@123, Content: Sample	File created successfully	Pass
3.	Create file with empty content	BVA	Filename: TestFile, Content: ``	File created successfully	Pass

4.	Fail to create file with null content	ECP	Filename: TestFile, Content: null	File creation fails	Pass
5.	Update file with valid input	ECP	File ID: 1, Name: UpdatedFile, Content: Updated	Update succeeds	Pass
6.	Fail to update non-existent file ID	ECP	File ID: -1, Name: UpdatedFile	Update fails	Pass
7.	Update with empty filename	BVA	File ID: 1, Name: , Content: Update	Update succeeds	Pass
8.	Fail to update with zero page number	BVA	Page Number: 0	Update fails	Pass
9.	Update with empty content	BVA	Content:	Update succeeds	Pass
10.	Update with null content	ECP	Content: null	Update succeeds	Pass
11.	Delete file with valid ID	ECP	File ID: 1	Delete succeeds	Pass
12.	Fail transliteration for non-existent page ID	ECP	Page ID: -1, Text: مرحبا	Returns null	Pass
13.	Fail transliteration for empty Arabic text	BVA	Page ID: 1, Text: ``	Returns null	Pass
14.	Transliterate very long Arabic text	BVA	Text: مرحبا repeated 1024 times	Transliteration succeeds	Pass
15.	Create file with very long filename	BVA	Filename: A repeated 255 times	File created successfully	Pass
16.	Delete file after creation	ECP	Create TestFileToDelete, then delete	File successfully deleted	Pass
17.	Retrieve files after creation	ECP	Create FileForGetTest, retrieve	File successfully retrieved	Pass

18.	Change file content in database	ECP	File ID: 1, Update content	File content updated successfully	Pass
19.	Fail to update file with non-existent ID	ECP	File ID: 9999	Update fails	Pass
20.	Fail to delete with non-existent ID	ECP	File ID: -1	Delete fails	Pass
21.	Create file with very large content	BVA	Content: A repeated 1 MB	File created successfully	Pass
22.	Create file with valid input	ECP	Filename, Content from Excel	Matches expected Excel result	Pass
23.	Create file with empty filename	BVA	Filename: , Content from Excel	Matches expected Excel result	Pass
24.	Transliterate file content correctly	ECP	File ID, Arabic text from Excel	Matches expected Excel result	Pass
25.	Create file with normal input	ECP	Filename: TestFile, Content: Sample	File created successfully	Pass
26.	Create file with empty content	BVA	Filename: EmptyContentFile, Content: ``	File created successfully	Pass
27.	Create file with special characters in name	ECP	Filename: Special@File#, Content: Sample	File created successfully	Pass
28.	Create file with large content	BVA	Content: 10,000 repetitions of large_content	File created successfully	Pass
29.	Update file with normal input	ECP	File ID, Updated name and content	File updated successfully	Pass
30.	Fail to update file with nonexistent page number	ECP	Page Number: 999	Update fails	Pass
31.	Update file with empty content	BVA	Content: ``	File updated successfully	Pass

32.	Fail to update with nonexistent file ID	ECP	File ID: 9999	Update fails	Pass
33.	Delete existing file	ECP	File ID exists	File deleted successfully	Pass
34.	Fail to delete nonexistent file	ECP	File ID: 9999	Delete fails	Pass
35.	Import text file with supported extension	ECP	.txt file	Import succeeds	Pass
36.	Fail to import file with unsupported extension	ECP	.pdf file	Import fails	Pass
37.	Get existing file	ECP	File ID exists	File retrieved successfully	Pass
38.	Fail to get nonexistent file	ECP	File ID: 9999	File retrieval fails	Pass
39.	Get file extension with extension	ECP	Filename: file.txt	Returns txt	Pass
40.	Get file extension with no extension	ECP	Filename: file	Returns empty string	Pass
41.	Get file extension with .md5	ECP	Filename: check.md5	Returns md5	Pass
42.	Get all files when no files exist	ECP	No files in DB	Returns empty list	Pass
43.	Get all files when multiple files exist	ECP	Two files in DB	Returns list with 2 files	Pass
44.	Transliterate normal Arabic text	ECP	Text: مرحبا	Returns transliterated text	Pass
45.	Transliterate empty Arabic text	BVA	Text: ``	Returns empty string	Pass
46.	Fail transliteration for nonexistent page ID	ECP	Page ID: 9999	Returns null	Pass

Running Tests

Run all tests using either Junit or XML file (Test NG):

Technology Stack

- **Frontend:** Swing
- **Backend:** Java
- **Database:** MariaDB
- **Testing Frameworks:** Junit, TestNG