STP Document

Mohammed Hussien

Table of Contents

Planned Schedule

A step in the project process	Start date	End date
Preparation of STP document	24.02.2024	25.02.2024
Preparation of STD document	26.02.2024	27.02.2024
Round of tests #1	28.02.2024	29.02.2024
Round of tests #2	29.02.2024	01.03.2024
Round of tests #3	01.03.2024	02.03.2024
Preparation of STR document	02.03.2024	03.03.2024

Purpose Of This Document

The purpose of the document is to define a framework program for testing This framework plan will include all relevant topics for planning and performing the tests, such as the test topics and types of tests, the schedule, and the planned work method for testing the system.

The document will also be used as base for writing a detailed test plan (STD) in which the various tests will be detailed step by step. After that, the tests will be carried out according to the instructions in this document.

Description Of the GeeksforGeeks System

GeeksforGeeks is a renowned online platform serving as a vibrant community for programmers, developers, and technology enthusiasts worldwide. Its core mission revolves around fostering a collaborative environment where individuals can access high-quality educational content, engage in discussions, and enhance their skills in various domains of computer science and programming. GeeksforGeeks has emerged as a go-to destination for both beginners and experienced professionals looking to expand their knowledge and expertise in areas such as data structures, algorithms, programming languages, system design, and more.

Purpose: GeeksforGeeks' primary objective is to democratize technical education and empower learners to excel in the field of computer science and programming. By offering a vast repository of tutorials, articles, coding challenges, and interview preparation resources, the platform aims to cater to the diverse learning needs of its audience, including students, educators, and industry professionals. GeeksforGeeks strives to facilitate continuous learning and skill development, equipping individuals with the tools and knowledge necessary to thrive in a rapidly evolving technological landscape.

Through its user-friendly interface and community-driven approach, the platform fosters collaboration, innovation, and lifelong learning among enthusiasts passionate about technology.

Key Features and Functions:

1. **Course Creation**: GeeksforGeeks provides a platform for instructors to create and publish courses covering a wide array of topics in computer science and programming, catering to varying skill levels and interests.
2. **Learning and Access**: Users can access a vast repository of educational content, including tutorials, articles, coding challenges, and practice exercises, available on-demand for self-paced learning.
3. **Enrollment and Updates**: Students can enroll in courses of their choice and receive notifications regarding course updates, new additions, or relevant announcements from instructors.
4. **Interaction and Engagement**: GeeksforGeeks facilitates interaction between instructors and learners through discussion forums, coding contests, and interactive coding environments, enhancing engagement and collaboration.
5. **Live Sessions and Webinars**: GeeksforGeeks hosts live Q&A sessions, webinars, and workshops conducted by industry experts and experienced practitioners, allowing learners to interact directly and gain insights in real-time.
6. **Progress Tracking and Analytics**: The platform offers tools for instructors to track student progress, assess performance, and analyze engagement metrics, enabling data-driven insights for course improvement.
7. **Community Support**: GeeksforGeeks fosters a vibrant and supportive community where learners can seek guidance, share knowledge, and collaborate with peers and experts in the field of computer science and programming.

Stakeholders:

1. **Course Instructors**: GeeksforGeeks is open to individuals, professionals, and institutions who create and publish courses to share their knowledge in computer science and programming.
2. **Students**: GeeksforGeeks is for individuals looking to enhance their skills in various technical areas through access to a wide range of educational content.
3. **Affiliate Marketers**: GeeksforGeeks provides opportunities for affiliates to earn commissions by promoting courses and resources available on the platform.
4. **Educational Institutions**: GeeksforGeeks offers supplementary learning resources for schools, colleges, and universities to integrate into their curriculum or provide additional support to students.

Glossary and Abbreviations

Glossary

- GUI (Graphical User Interface): The design of user interfaces based on specified requirements.
- Functional Testing: Verification that fundamental system functions operate correctly.
- Maintenance Testing: Examination of the functionality of a modified system following changes, updates, or alterations in the working environment.
- STP (System Test Plan): A comprehensive project planning document encompassing strategy, schedule, and topic tree.
- STD (System Test Design): Detailed documentation outlining the testing plan.
- STR (System Test Results): A concise document summarizing test results after three cycles.
- Traceability Matrix: A document that correlates any two baselined documents that require a many-to-many relationship to determine the completeness of the relationship.

Abbreviations

- QA: Quality Assurance
- CEO: Chief Executive Officer
- HR: Human Resources

Testing Plan

The testing team will communicate the requirements to the development team based on the planned tests. The development team will work according to these requirements to maximize efficiency.

Resources: Team leader, 4 testers. A total of 5 team members will be assigned to this project.

i

1. **Computers:**
    - High-performance desktops or laptops to run test scripts and perform manual testing.
2. **Network Equipment:**
    - Routers and switches to simulate different network conditions for testing YouTube's performance under various network scenarios.
3. **Mobile Devices:**
    - Various smartphones and tablets to test YouTube's mobile app across different platforms (iOS, Android).
4. **Webcams and Microphones:**
    - Testing features like video uploads, live streaming, and video conferencing may require webcams and microphones.
5. **Virtual Machines:**
    - Set up virtual machines for testing on different operating systems and browser combinations.
6. **Storage Devices:**
    - Sufficient storage space to store test data, logs, and video files generated during testing.

Software Needed:

1. **Operating Systems:**
    - Install and configure various operating systems for testing, including Windows, macOS, and Linux distributions.
2. **Browsers:**

- Latest versions of popular browsers (Google Chrome, Microsoft Edge, Safari) for cross-browser compatibility testing.

3. **Mobile Emulators/Simulators:**
    - Emulators or simulators to test GeeksforGeeks mobile website on different devices and screen sizes.
4. **Performance Testing Tools:**
    - Tools like Apache JMeter, LoadRunner, or Gatling for performance and load testing to simulate high user traffic.
5. **Automation Testing Tools:**
    - Selenium, Appium, or similar tools for automating functional tests across web and mobile platforms.
6. **Database Management System:**
    - Database systems (MySQL, PostgreSQL) for testing GeeksforGeeks data handling and retrieval functionality.
7. **Collaboration Tools:**
    - Communication and collaboration tools (e.g., Slack, Microsoft Teams) for effective communication among the testing team.
8. **Test Management Tools:**
    - Test case management tools (e.g., **TestRail, Jira**) to organize, execute, and track test cases.

This document will be approved by the testing team leader and the project manager. Following approval, the senior tester will build the STD document based on it. The STD document will then undergo approval by the testing team leader.

After approval of the STD document, three rounds of testing will be conducted. At the end of these rounds, the STR document will be prepared by the senior tester, undergo approval by the testing team leader, and serve as the final documentation.

Before the start of testing rounds, **functional tests** will be performed, including:

1) **Unit Testing:** To test individual units or components of a software application.
2) **Sanity Testing:** To verify that the most important functionalities of a software application work correctly.
3) **Integration Testing:** To verify the interactions and interfaces between different components or systems within the application.
4) **Regression Testing:** To ensure that new code or changes do not affect the existing functionality of the software.
5) **API Testing:** To validate the functionality of an API by testing its endpoints and request-response mechanisms.
6) **UI Testing:** To validate that the user interface elements and interactions function correctly according to the design.
7) **End-to-End Testing:** To evaluate the entire software system's functionality from start to finish, simulating real user scenarios and interactions.

Afterwards, the following **non-functional tests** will be conducted including:

1) **Performance Tests (Load + Stress + Volume):** they assess how well a system performs under various conditions, including heavy loads, stressful situations, and large data volumes.
2) **Security Tests:** Security tests aim to identify vulnerabilities and weaknesses in a system to ensure protection against unauthorized access.
3) **Upgrade and Installation Test:** Upgrade and installation tests verify the smooth installation of software upgrades and updates, ensuring that the system remains stable and functional.
4) **Recovery Tests:** they assess how well a system can recover from failures including data loss or system crashes.
5) **Localization and Globalization Testing:** Localization testing verifies that a software application adapts to specific regional or cultural requirements, while globalization testing ensures its compatibility with diverse international settings.
6) **Usability Testing:** evaluates the user friendliness and overall user experience of a software application to ensure it meets user expectations.
7) **Compatibility Testing:** Compatibility testing ensures that a software application works seamlessly across different devices, browsers, operating systems.

Starting and exiting Criteria

- **Criteria for starting the tests:**

- 100% of the planned sanity tests were carried out and passed successfully.
- 100% of planned functional test cases have been created and reviewed.
- A traceability matrix is established, linking each test case to specific requirements.
- The testing environment, including necessary configurations, data, and tools, is prepared and verified.
- Sufficient and accurate test data for both positive and negative scenarios is available.
- The test plan, detailing the testing approach, objectives, and schedules, has been reviewed and approved.

- **Completion/Release Criteria:**

- 100% of planned functional and non-functional tests have been executed, and results have been documented.
- 100% of test cases passed successfully.
- All critical bugs have been fixed at this point.
- The remaining bugs are at low severity levels, with no high-severity issues affecting functionality.

Traceability Table

Business Requirement	REQ ID	Functional/non Functional tests	Test Case ID	Defects?	Status
Sign-In	1	sign in to GeeksforGeeks account.	1.a	None	Verified
		Unsuccessful sign in to GeeksforGeeks account	1.b	None	Verified
		Forgot password	1.c	None	Verified
		Perform regular GeeksforGeeks sign-in under unstable network conditions.	1.d	None	In-progress
		multiple users attempt to log in simultaneously on GeeksforGeeks	1.e	None	Denied
Sign out	2	Successful sign out from the user account	2.a	None	Verified
		Verify that the user data is still in the database after signing out	2.b	None	Verified
Sign-Up	3	Successful sign up on GeeksforGeeks	3.a	None	Verified
		Unsuccessful sign up on GeeksforGeeks	3.b	None	Verified
		Simulate a server outage during the sign-up process and assess the system's recovery.	3.c	None	In-progress
Searching	4	Basic Search Functionality	4.a	None	Verified
		Verify Search suggestion	4.b	None	Verified
		Searching in a foreign language	4.c	None	Verified
		Searching invalid chars	4.d	None	Verified
Articles	**5**	Verify that a use can successfully create an article and publish it	5.a	None	Verified
		Verify that a user can edit an article name	5.b	None	Verified
		Verify that a user can edit the article content	5.c	None	Verified
		Verify that different languages can be included in the article	5.d	None	Verified
Videos	**6**	Verify successful video playing	6.a	None	Verified

		Verify successful video saving	6.b	None	Verified
		Verify successful video unsaving.	6.c	None	Verified
Run code	**7**	Verify that a user can successfully run a code from the website	7.a	None	Verified
		Verify that a user can run codes in different browsers	7.b	None	Verified

Testing Tree

1. Sign- In/out:

Functional testing:

a. sign in to GeeksforGeeks account.

i. Verify successful login with a valid email and password.

ii. Verify unsuccessful login with an invalid email or password.

1. Attempt login with an invalid email.
2. Attempt login with a valid email and incorrect password.

iii. forgot password:

1. retrieve password using email address.
2. retrieve password using phone number.

b. Sign up for a GeeksforGeeks account.

i. Successful sign out from the user account

ii. verify that the sign out button is present and clickable.

Non-functional:

Reliability Testing

c. Perform regular GeeksforGeeks sign-in under unstable network conditions.

d. Verify that the user data is still in the database after signing out .

Scalability Testing

e. Test the sign-in functionality with a large number of concurrent users:

1. multiple users attempt to log in simultaneously on GeeksforGeeks.

2. Sign-Up:

Functional testing:

a. signs up to GeeksforGeeks using email address.

i. Verify successful registration with a valid email and password.

ii. Verify unsuccessful registration with an invalid email or password.

1. Attempt registration with an already used email.
2. Attempt registration with a fake email.
3. Attempt registration with a weak password.

Non-functional:

Recovery Tests:

b. Simulate a server outage during the sign-up process and assess the system's recovery.

Localization and Globalization Testing:

c. Perform sign-up using different regional settings (language, date formats).

3.Searching

Functional Testing

a. Basic Search Functionality:

i. The search bar is present and functional.

ii. successfully Performing basic search in different languages.

iii. verify that we get no result when searching for invalid unputs.

b. Advanced Search Options:

i. successfully Applying date filter before searching.

ii. Verify sorting by relevance.

c. Search Suggestions:

i. Confirm suggestions while typing something on the search bar.

ii. Verify the relevance of suggestions.

Non-Functional:

Scalability Testing:

d. Simulate a scenario where a significantly large number of users are simultaneously conducting search operations on GeeksforGeeks.

Performance Testing:

e. Test the performance of applying filters and sorting options during searches.

Compatibility Testing

f. Test search functionality across different browsers.

g. Validate responsiveness on various devices (desktop, mobile).

Usability Testing

h. Make sure the search engine is placed in a way that is understandable and clear to the user.

4. Articles:

Functional testing:

a. article creating/modifying:

i. Verify successful article creation.
ii. Verify unsuccessful registration with an invalid article name.
iii. Verify successful article change name.
iiii. Verify successfully creating article with foreign languages.

Non-functional:

Localization and Globalization Testing:

b. Perform Article-Writing using different regional settings (language, date formats).

Reliability Testing:

c. Verify that articles are consistently saved and published without loss of data, even during system failures or interruptions.

Security Testing:

d. Test the system for vulnerabilities related to unauthorized access or manipulation of articles.

5. Videos:

Functional testing:

a. videos logic:

i. Verify successful video playing.
ii. Verify successful video saving.
iii. Verify successful video unsaving.
iiii. Verify successful video commenting.

Non-functional:

performance Testing:

b. Test the loading time of videos under different network conditions to ensure optimal performance.

c. Measure the system's response time when users interact with video-related features like saving or unsaving videos.

Compatibility Testing:

d. Verify that saved videos are accessible and playable across different platforms without any compatibility issues.

Security Testing:

e. Test for vulnerabilities related to unauthorized access or manipulation of videos.

5. code run:

Functional testing:

a. code run logic:

i. Verify successful execution of code.

ii. Verify successful presence of the code running button.

iii. Verify successful comparison of expected vs. actual output.

iiii. Verify successful validation of code syntax.

Non-functional:

performance Testing:

b. Measure the time taken for the code to execute and display the result, ensuring acceptable performance for users.

c. Measure the system's response time when users interact with video-related features like saving or unsaving videos.

Reliability Testing:

d. Verify that code execution is consistent and reliable, with accurate results provided for all inputs.

Scalability Testing:

e. Test the system's ability to handle multiple concurrent code execution requests from different users.

Hazard Table

Responsible	Description	Action	Hazard Description	Risk Level	Damage	Chance	Hazard	#
	-Description Of prevention method	Enclose Monitoring Acceptance	What will happen in case of the hazard	Chance Multiplied by Damage	10-1	The probability of the hazard occurring, ranging from 0 to 1.		
System	NA	Monitoring	Unable to connect as a user	5	10	0.5	Bad Internet Connectivity	**1**
	Finding a tester for the project length	Enclose	Bad testing and coverage	1.6	8	0.2	New Testers	**2**
System		Monitoring	Unable to login and retrieve info from the DB	7	7	0.5	Server Crash	**3**
	Postponing / finding replacement	Enclose	Vacations	7	7	1.0	Vacations	**4**
System	Adding servers	Enclose	System Crash	5	10	0.5	Weak Server	**5**

HR	Hiring a stable worker	Monitoring	Lowering Team Morale	5	10	0.5	Employee quitting	**6**
QA Lead	Hiring Experienced Testers	Enclose	Failure to meet the schedule	1	10	0.1	Inexperienced Testers	**7**
CEO	More flexible customer	Monitoring	No income for the company	3	10	0.3	Contract Termination	**8**
CEO	QA Lead bad Management	Enclose	Unsatisfied Customer	10	10	1	Failure to meet the schedule	**9**
CEO	An appointment must be made with the customer and it should be noted to him that it will not be possible to make changes after the system is established	Monitoring	Failure to be prepared for changes by the customer will not ensure a professional, accurate and correct inspection	3.5	7	0.5	Customer requirements document changes frequently during the project	**10**