LAB-6

Introduction to Software Testing (ST):

Software Testing (ST) is the process of evaluating and verifying that a software application meets its requirements and functions correctly. It helps identify defects, ensures software reliability, and enhances security, performance, and usability.

Why is Software Testing Important?

✔ **Ensures Software Quality** – Detects and fixes defects early, ensuring a stable and reliable system.
✔ **Enhances Security** – Protects sensitive stock market data from unauthorized access and cyber threats.
✔ **Improves Performance** – Ensures smooth operation under normal and extreme market conditions.
✔ **Reduces Costs** – Early detection of issues lowers development and maintenance expenses.
✔ **Increases User Satisfaction** – Provides a smooth, error-free experience for traders and analysts.

Types of Testing

1. Sanity Testing:

Sanity Testing is a quick check after minor changes or bug fixes to ensure that the core functionalities work as expected. It is not an exhaustive test but ensures no major issues exist before further testing.
📌 **Example:** If the developer fixes a bug in the stock prediction model, sanity testing ensures the model still runs correctly without checking all other functionalities.

2. Security Testing:

Security Testing identifies vulnerabilities and ensures that the system protects sensitive stock and user data.
📌 **Example:** Testing if unauthorized users can access historical stock data meant only for premium users.

3. Stress Testing:

Stress Testing examines how the system performs under extreme conditions, such as handling a massive number of stock queries or processing large datasets.

📌 **Example:** Checking if the system crashes when 10,000 traders access stock predictions at the same time.

4. Performance Testing:

Performance Testing measures system responsiveness, speed, and stability under expected workloads.
📌 **Example:** Ensuring stock trend analysis results load within 3 seconds during normal operations.

Test Cases for Each Testing Category

1. Sanity Testing:

Sl. No.	Test Case ID	Test Input	Processing	Expected Output
1	ST_01	Login with valid credentials	User enters username & password	User successfully logs in
2	ST_02	Open stock trend dashboard	System fetches data	Dashboard displays correctly
3	ST_03	Fetch latest stock prices	System retrieves real-time prices	Updated stock data shown
4	ST_04	Generate a stock market report	System processes report request	Report is generated successfully
5	ST_05	Logout from the system	User clicks logout	User is redirected to login page

2. Security Testing:

Sl. No.	Test Case ID	Test Input	Processing	Expected Output
1	SEC_01	SQL Injection attack on login page	System processes input	Login should be denied
2	SEC_02	Check encryption of stored user data	System verifies storage method	Data should be securely encrypted
3	SEC_03	Attempt to access premium stock data without authorization	Unauthorized user requests access	Access should be denied
4	SEC_04	Stay inactive for 15 minutes	System monitors session activity	Auto logout should occur
5	SEC_05	Simulate brute force attack on login	Multiple incorrect attempts	Account temporarily locked

3. Stress Testing:

Sl. No.	Test Case ID	Test Input	Processing	Expected Output
1	STRESS_01	10,000 users accessing stock predictions simultaneously	System processes high traffic	System remains stable
2	STRESS_02	Load 1 million stock records	System handles bulk data	No crashes or slowdowns
3	STRESS_03	500 users request API data at the same time	System processes API calls	No downtime should occur
4	STRESS_04	System running with 90% CPU load	Monitor system performance	No major slowdown occurs
5	STRESS_05	Open 100 stock charts simultaneously	System loads multiple charts	System handles without freezing

4. Performance Testing:

Sl. No.	Test Case ID	Test Input	Processing	Expected Output
1	PERF_01	Login with valid credentials	Measure response time	Response < 2 sec
2	PERF_02	Load stock analysis report	System compiles stock data	Report loads in < 3 sec
3	PERF_03	Handle 1,000 API requests per minute	System processes API calls	Response time < 1 sec
4	PERF_04	Display real-time stock trends	System fetches and displays stock prices	Data updates within 1 sec
5	PERF_05	Search for a stock in a 1M-record database	Run search query	Results displayed in < 2 sec

Conclusion

This exercise helps in understanding and applying different software testing methodologies to ensure that the **Stock Market Prediction System** is **functional, secure, stable, and high-performing**. These test cases ensure that the system delivers accurate stock predictions, maintains security, and handles high user loads effectively.