High-Level Regression Test Plan for APIs

1. Objectives

- Ensure that existing functionality remains unaffected by new changes or enhancements.
- Validate the stability and reliability of APIs after patches, upgrades, or code changes.

---

2. Scope

- All existing APIs related to the system, including public, private, and third-party integrations.
- Key functional areas: Authentication, Data Processing, Business Logic, Error Handling, and Integrations.

---

3. Test Strategy

1. **Test Case Review and Prioritization**
    - **High Priority**: Core functionalities, critical business workflows, and frequently used APIs.
    - **Medium Priority**: APIs with moderate impact on functionality.
    - **Low Priority**: Less critical APIs with limited user interaction or dependencies.
2. **Test Automation**
    - Automate regression test cases using frameworks like Postman, Rest Assured, Junit, Test Ng, Java, Playwright etc for consistency and speed.
    - Use CI/CD pipelines (e.g., Jenkins, Azure DevOps) for automated regression test execution after each deployment.
3. **Environment Setup**
    - Ensure parity between staging and production environments **(Active-Active).**
    - Mock third-party APIs if unavailable or unstable during testing.

---

4. Key Activities

1. **Test Data Preparation**
    - Generate or reuse data sets to cover all API scenarios, including valid, invalid, and edge cases.

2. **Functional Testing**
    - Validate API inputs and outputs against requirements (e.g., response codes, payload structure, and business rules).
3. **Integration Testing**
    - Check interactions between APIs, ensuring proper data flow and communication.
4. **Non-Functional Testing**
    - Validate performance (response time, throughput) and reliability under load.
5. **Backward Compatibility Testing**
    - Ensure new changes do not break older API clients or existing integrations.
6. **Error Handling**
    - Test for proper error messages, status codes, and exception handling.
7. **Security Testing**
    - Validate authentication, authorization, data encryption, and protection against vulnerabilities like SQL injection or XSS.

---

5. Test Execution Plan

- **Frequency**: Post every sprint, patch, or major code merge.
- **Execution Types**:
    - Smoke Testing: Basic checks to ensure the system is ready for deeper testing.
    - Regression Testing: Full suite execution of API test cases.
- **Duration**: Should align with release timelines, with emphasis on optimizing for rapid execution.

---

6. Reporting and Tracking

- **Metrics**:
    - Test Coverage: Percentage of APIs tested.
    - Pass/Fail Rate: Success rate of regression tests.
    - Defects: Number, severity, and resolution time of identified issues.
- **Tools**: Use test management tools like Zephyr, TestRail, or JIRA for tracking and reporting.

---

7. Risk Management

- Prioritize testing APIs with the highest business impact.
- Regularly update regression suites to include new bug fixes or features.
- Maintain a rollback plan in case of critical failures.

---

8. Deliverables

- Updated regression test suite.
- Test execution reports with detailed logs.
- Bug reports and resolutions.