# Netflix Web Application Automation Test Plan

## 1. Introduction

This document outlines the test plan for the automation testing of the Netflix web application. The primary focus is on ensuring the accuracy and reliability of key functionalities, including login, account management, search, general navigation, and playback controls. The tools and frameworks used in this project include Python, Selenium, Jupyter Notebook, JIRA, and Excel for test case documentation. Cross-browser testing will be conducted on Chrome and Edge.

## 2. Objective

The objective of this test plan is to validate the proper functioning of essential Netflix web application features through automated testing. This will:  
- Ensure a seamless user experience for critical user actions.  
- Identify and report any defects effectively.  
- Provide comprehensive test coverage across supported browsers.

## 3. Scope

The following functionalities will be tested:  
- **Login**: Verifying user login/logout functionality with valid and invalid credentials.  
- **Account** **Management**: Testing user profile selection, deletion, and updates.  
- **Search**: Validating search functionality for movies, series, and genres.  
- **General** **Navigation**: Ensuring smooth navigation across home, movies, My List, etc.  
- **Playback** **Controls**: Testing video playback, pause, forward, rewind, and volume controls.

## 4. Test Environment

- **Operating** **Systems**: Windows 11.  
- **Browsers**: Chrome (latest version), Edge (latest version).  
- **Automation** **Tools**: Python, Selenium WebDriver.  
- **IDE/Platform**: Jupyter Notebook.  
- **Bug** **Tracking** **Tool**: JIRA.  
- **Documentation** **Tools**: Excel for test case management and Word for other deliverables.

## 5. Test Deliverables

- Test Plan Document  
- Automated Test Scripts  
- Test Case Document (Excel sheet)  
- Test Execution Report  
- Defect Report (via JIRA)

## 6. Approach

1. **Test** **Design**:  
 - Identify test scenarios based on user stories and requirements.  
 - Design detailed test cases in Excel for each identified scenario.  
  
2. **Test** **Automation**:  
 - Write automation scripts using Python and Selenium.  
 - Implement reusable functions for common tasks like login and navigation.  
  
3. **Test** **Execution**:  
 - Execute test scripts on Chrome and Edge browsers.  
 - Log defects in JIRA with detailed steps and screenshots.  
  
4. **Defect** **Management**:  
 - Categorize and prioritize defects.  
 - Collaborate with developers for resolution.  
  
5. **Reporting**:  
 - Prepare a test execution report summarizing passed, failed, and skipped test cases.

## 7. Schedule

|  |  |  |
| --- | --- | --- |
| Activity | Start Date | End Date |
| Test Case Design | 2024-12-26 | 2024-12-28 |
| Script Development | 2024-12-29 | 2024-12-31 |
| Test Execution | 2025-01-01 | 2025-01-03 |
| Defect Logging & Retest | 2025-01-04 | 2025-01-05 |
| Reporting | 2025-01-06 | 2025-01-07 |

## 8. Risks and Assumptions

**Risks**:  
- Browser-specific issues during playback testing.  
- Potential delays in script development.  
  
**Assumptions**:  
- Test data and user accounts are pre-configured.  
- Browsers are updated to the latest versions.

## 9. Exit Criteria

Testing will conclude when:  
- All critical and high-severity defects are resolved.  
- Automated test scripts are executed successfully on both browsers.  
- Test execution report is reviewed and approved.

## 10. Conclusion

This test plan provides a comprehensive framework for the automation testing of the Netflix web application’s critical functionalities. By following this plan, the project aims to deliver a robust, defect-free user experience across supported browsers.