Verbatimly App Summarized Test-Plan

PREPARED BY: Theophilus Onolunose

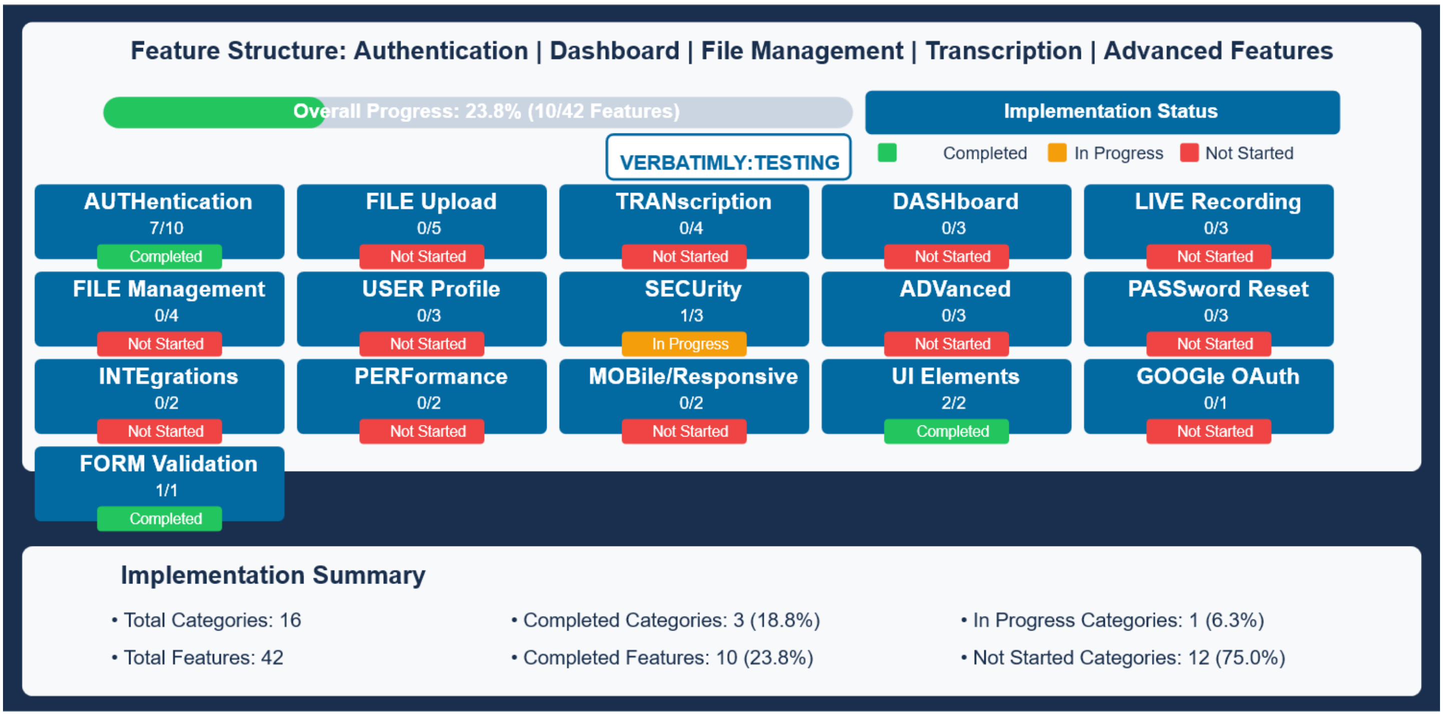
DATE: 28.07.2025

1. Overview

As part of my project initiative, I am motivated to execute a comprehensive automated testing regimen for the web application located at 'https://verbatimly.com.

The product owner is ' Tech von chu' at Kolpingstr. 83, 53359, Rheinbach, Germany.

Overview of the Features to be Tested (Test items) and the Current Test Coverage:



Feature Structure: Authentication | Dashboard | File Management | Transcription | Advanced Features

Overall Progress: 23.8% (10/42 Features)

Implementation Status

🟩 Completed 🟧 In Progress 🟥 Not Started

VERBATIMLY:TESTING

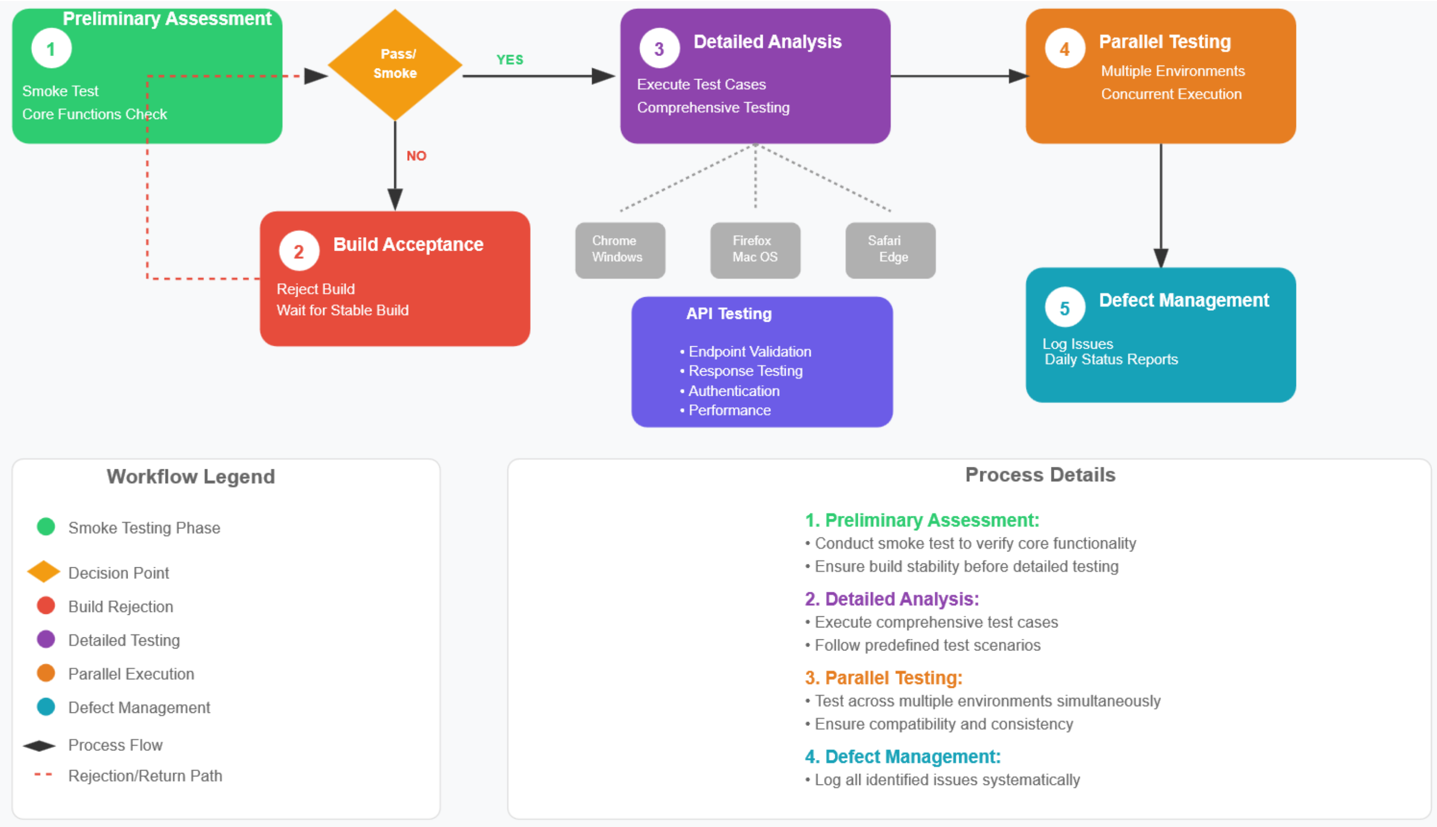
AUTHentication 7/10 Completed	**FILE Upload** 0/5 Not Started	**TRANscription** 0/4 Not Started	**DASHboard** 0/3 Not Started	**LIVE Recording** 0/3 Not Started
FILE Management 0/4 Not Started	**USER Profile** 0/3 Not Started	**SECUrity** 1/3 In Progress	**ADVanced** 0/3 Not Started	**PASSword Reset** 0/3 Not Started
INTEgrations 0/2 Not Started	**PERFormance** 0/2 Not Started	**MOBile/Responsive** 0/2 Not Started	**UI Elements** 2/2 Completed	**GOOGle OAuth** 0/1 Not Started
FORM Validation 1/1 Completed				

Implementation Summary

- Total Categories: 16
- Completed Categories: 3 (18.8%)
- In Progress Categories: 1 (6.3%)
- Total Features: 42
- Completed Features: 10 (23.8%)
- Not Started Categories: 12 (75.0%)

The overview of the structured test workflow is presented in the figure below:

2. Structured Test Workflow



Workflow Legend

- ● Smoke Testing Phase
- ◆ Decision Point
- ● Build Rejection
- ● Detailed Testing
- ● Parallel Execution
- ● Defect Management
- ⟷ Process Flow
- - - Rejection/Return Path

Process Details

1. Preliminary Assessment:
- Conduct smoke test to verify core functionality
- Ensure build stability before detailed testing

2. Detailed Analysis:
- Execute comprehensive test cases
- Follow predefined test scenarios

3. Parallel Testing:
- Test across multiple environments simultaneously
- Ensure compatibility and consistency

4. Defect Management:
- Log all identified issues systematically

3. Test Environments

- Windows 11 -- Chrome, Firefox, and Edge
- Mac OS -- Safari Browser

4. Test Strategy

With the application's user interface and functionality provided by 'Verbatimly' in their online platform, I have discerned the necessity for a comprehensive Functional Testing regimen tailored to the functionalities delineate in the "Test items" section. The strategic approach to this testing initiative encompasses:

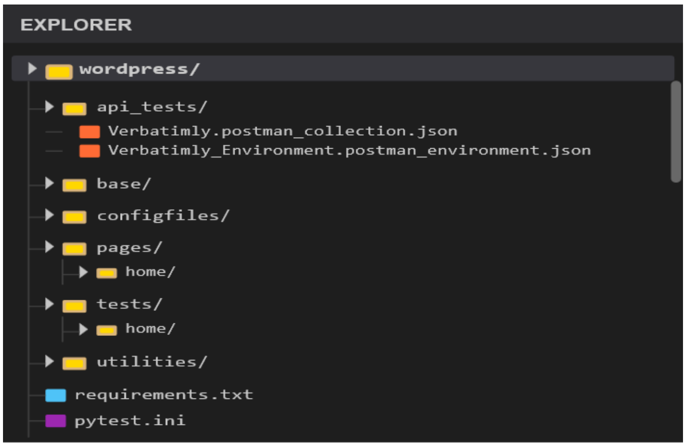
- Test Design: Systematic development of test scenarios that encapsulate the diverse functionalities within the testing scope.
- Test Case Formulation: Rigorous creation of detailed test cases derived from the aforementioned scenarios, ensuring a holistic evaluation of each feature. Though the test cases adaptation is still in progress.
- Execution Strategy: Leveraging advanced automated testing tools and methodologies to ensure precision and efficiency in the testing efforts.

5. Key Testing Tools and Management:

- Selenium **WebDriver**: Automates browser actions to validate end-to-end web application functionality and it is a free source tool.
- Pytest **Framework**: Provides structured test execution with fixtures, assertions, and parameterization for maintainable automation.
- Postman **for API**: Tests REST APIs, validates responses, and automates API workflows with collection

6. Verbatimly Test Automation Framework

1. Project Structure



```
EXPLORER
▶ wordpress/
  ▶ api_tests/
    Verbatimly.postman_collection.json
    Verbatimly_Environment.postman_environment.json
  ▶ base/
  ▶ configfiles/
  ▶ pages/
    ▶ home/
  ▶ tests/
    ▶ home/
  ▶ utilities/
  requirements.txt
  pytest.ini
```

2. Prerequisites

- pip install -r requirements.txt

3. Run API Tests (Postman via Newman)

```
cd wordpress
newman run api_tests/Verbatimly.postman_collection.json \
  -e api_tests/Verbatimly_Environment.postman_environment.json
```

4. Run Selenium UI Tests (Pytest)

Login Tests

```
pytest tests/home/login_tests.py -v --tb=short --disable-warnings -s
```

Create Account Tests

```
pytest tests/home/createAccount_tests.py -v --tb=short --disable-warnings -s
```

5. Logs & Reports

- Execution logs: automation.log
- Screenshot: Screenshot shows failed test screenshot