



# **Verbatimly App Summarized Test-Plan**



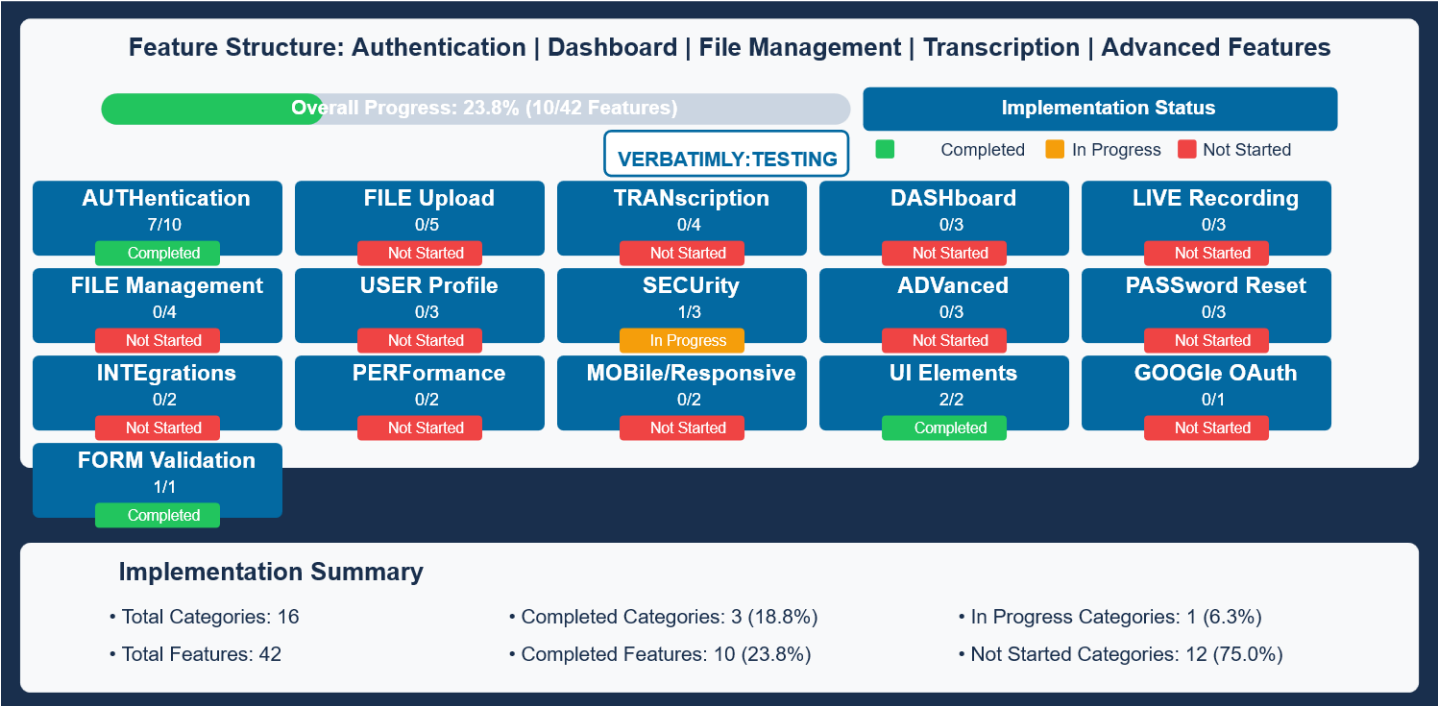
**PREPARED BY: Theophilus Onolunose**  
**DATE: 28.07.2025**

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://verbatimimly.com'.

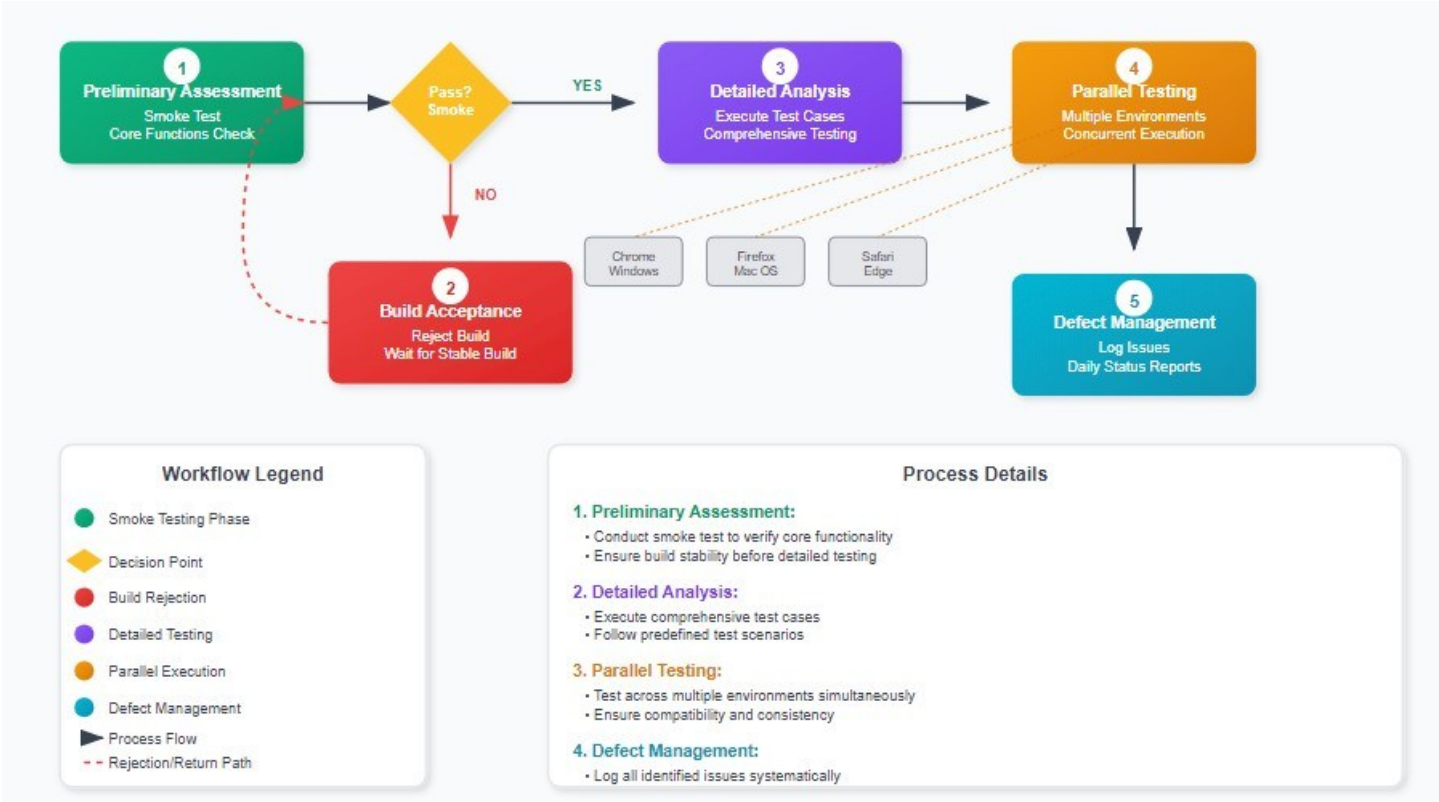
The product owner is 'Tech von chu' at Kolpingstr. 83, 53359, Rheinbach, Germany.

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



The UI structured test overview is presented in the figure below:

2. Structured Test Workflow



### 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.
- **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

```
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`

#### 4. Run API Tests (Postman via Newman)

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

#### 5. 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
```

#### 6. Logs & Reports

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