Manual Test Cases

The following section outlines manual test cases designed to verify the functional correctness of the PulsePredict system. These tests validate the core functionality of each stage in the audio-to-prediction pipeline, ensuring expected system behaviour for transcription, entity extraction, adverse event labelling, feature engineering, and final model prediction. Each test case includes a scenario description, step-by-step procedure, necessary prerequisites, test data, and expected vs. actual results.

Test Case ID: #PP001

Test Scenario: To verify successful transcription of an audio file using Whisper.

Test Steps:

- The user places a valid .mp3 or .wav audio file in the audio/ folder.
- The user runs the predict_from_audio.py script.
- The system initializes the Whisper model for transcription.
- The transcription process completes.
- A .txt file is saved in the transcripts/ folder.

Prerequisites:

The user must have a valid audio file of a doctor-patient conversation and a functioning Python environment with Whisper installed.

Browser: Not applicable (CLI-based process)

Device: Windows 11 PC with Python 3.10 installed

Test Data:
Valid .mp3 file (e.g., sample_consultation.mp3) with clear medical dialogue.

Expected/Intended Results:

A corresponding .txt transcription file is generated in the transcripts/ folder containing readable and complete dialogue.

Actual Results:

As expected.

Test Status – Pass/Fail:

Pass

Test Case ID: #PP002

Test Scenario: To verify successful extraction of medical entities using AWS Comprehend Medical.

Test Steps:

- The user ensures a transcript is available in the transcripts/ folder.
- The user runs the entity extraction module or full pipeline.
- The system sends text to AWS Comprehend Medical.
- Entities are returned and saved to the entities/ folder.

Prerequisites:

AWS credentials must be configured, and the input transcript must exist.

Browser: Not applicable

Device: Windows 11 PC

Test Data:

Transcript file containing text such as “The patient reports headaches and is prescribed ibuprofen.”

Expected/Intended Results:

A list of medical entities (e.g., symptoms, medications) is extracted and saved to entities/.

Actual Results:

As expected.

Test Status – Pass/Fail:

Pass

Test Case ID: #PP003

Test Scenario: To verify correct labelling of entities using FAERS adverse event list.

Test Steps:

- The user confirms an entity file exists in entities/.
- The user runs the labelling module.
- The system compares extracted entities with FAERS dataset.
- Labeled results are stored in labeled_entities/.

Prerequisites:

FAERS dataset must be loaded; entity extraction must be complete.

Browser: Not applicable

Device: Windows 11 PC

Test Data:

Entity list including keywords like “rash”, “fever”, “dizziness”.

Expected/Intended Results:

Entities are labeled as adverse or non-adverse and saved appropriately.

Actual Results:

As expected.

Test Status – Pass/Fail:

Pass

Test Case ID: #PP004

Test Scenario: To verify correct feature engineering from labeled entities.

Test Steps:

- The user ensures labeled entities are available.
- The user runs the feature engineering module.
- The system calculates features such as num_symptoms, num_adverse_events, adverse_event_ratio.
- The final feature set is saved as features.csv.

Prerequisites:

The labeled entity file must be ready.

Browser: Not applicable

Device: Windows 11 PC

Test Data:

Labeled data with multiple symptoms and at least one adverse event.

Expected/Intended Results:

features.csv is generated with all expected numeric fields for model input.

Actual Results:

As expected.

Test Status – Pass/Fail:

Pass

Test Case ID: #PP005

Test Scenario: To verify rule-based prediction logic based on adverse_event_ratio.

Test Steps:

- The user confirms features.csv is available.
- The user runs the evaluate_model.py script with rule-based logic.
- Predictions are generated based on a threshold value (e.g., 0.3).
- Results are printed or logged.

Prerequisites:

Feature data must exist; rule threshold logic must be defined.

Browser: Not applicable

Device: Windows 11 PC

Test Data:

Feature row with adverse_event_ratio = 0.4.

Expected/Intended Results:

The case is predicted as an adverse event (True).

Actual Results:

As expected.

Test Status – Pass/Fail:

Pass

Test Case ID: #PP006

Test Scenario: To verify that the trained ML model predicts adverse events correctly.

Test Steps:

- The user confirms the trained model exists in the model/ directory.
- The user runs predict_from_audio.py with valid audio input.
- The model processes features and makes a prediction.
- The prediction is displayed or logged.

Prerequisites:

Random Forest model must be trained and saved in the correct path.

Browser: Not applicable

Device: Windows 11 PC

Test Data:

Feature row with moderate to high adverse signal features.

Expected/Intended Results:

Prediction output is shown as either True or False indicating adverse event.

Actual Results:

As expected.

Test Status – Pass/Fail:

Pass

Test Case ID: #PP007

Test Scenario: To verify complete end-to-end pipeline from audio input to final prediction.

Test Steps:

- The user places an audio file in audio/.
- The user runs predict_from_audio.py.
- The system performs transcription, entity extraction, labeling, feature engineering, and prediction.
- Outputs are saved at each stage and a final prediction is returned.

Prerequisites:

All modules, models, and dependencies must be installed and configured.

Browser: Not applicable

Device: Windows 11 PC

Test Data:

Valid .mp3 file containing at least 30 seconds of a medical dialogue.

Expected/Intended Results:

Each intermediate step completes successfully and a final prediction is returned.

Actual Results:

As expected.

Test Status – Pass/Fail:

Pass

Test Case ID: #PP008

Test Scenario: To verify system behaviour when the audio file is missing or corrupted.

Test Steps:

- The user runs predict_from_audio.py without providing a valid audio file.
- The system attempts to load the file.
- An error is raised or logged gracefully.

Prerequisites:

No or invalid audio file is supplied.

Browser: Not applicable

Device: Windows 11 PC

Test Data:

Non-existent or corrupted file name like broken_audio.mp3.

Expected/Intended Results:

The system logs a clear error message and exits without crashing.

Actual Results:

As expected.

Test Status – Pass/Fail:

Pass