| **User Story ID** | **User Story** | **Given** | **When** | **Then** |
| --- | --- | --- | --- | --- |
| 1 | As an investigator, I can upload an image or video so that I can quickly check if it has been manipulated or AI-generated. | I have an image or video file | I upload the file to the system | I can see whether it has been flagged as manipulated or AI-generated |
| 2 | As an investigator, I can see clear visual indicators (heatmaps or highlighted frames) so that I can easily identify suspicious areas. | The system has analyzed the file | I view the results in the dashboard | I can see highlighted suspicious areas (heatmap or frames) |
| 3 | As an investigator, I can download a simple report in PDF format so that I can use it in my case files. | I have an analyzed result | I click “Export PDF” | A PDF report downloads including findings and summary |
| 4 | As a forensic analyst, I can review metadata and anomaly scores for deeper verification. | The file is uploaded | I open the “Detailed View” tab | Metadata and anomaly scores are displayed |
| 5 | As a forensic analyst, I can export results in JSON format to integrate into workflows. | I have results available | I click “Export JSON” | JSON file is downloaded with all detection data |
| 6 | As a forensic analyst, I can see frame-by-frame video analysis. | I uploaded a video | I select “Frame Analysis” | I can see suspicious frames flagged |
| 7 | As a legal officer, I want reports to include file hashes and chain-of-custody records. | A PDF report is generated | I open/download it | It includes SHA256 hash + chain-of-custody log |
| 8 | As a client, I want the system to work offline in secure environments. | The tool is installed locally | I start the application without internet | It functions fully in offline mode |
| 9 | As a client, I want reliable confidence scores with each detection. | I upload a file | I see detection results | Each result includes a percentage confidence score |
| 10 | As a product owner, I want to prioritize stories based on feedback. | Feedback is collected | I view backlog in dashboard | Stories are ranked by urgency/priority |
| 11 | As a developer, I want modular detection components for easy upgrades. | A new model is available | I replace detection module | System still works with new model |
| 12 | As a developer, I want logs of every analysis for debugging. | A file is analyzed | I open system logs | Logs show timestamp, model used, and result |
| 13 | As a journalist/public user, I want to verify authenticity of online videos easily. | I copy a video link or file | I submit it to system | I get a simple result: “Likely real” or “Likely AI-generated” |

**AC01 – Upload image/video for manipulation check**

* **User Story**: As an investigator, I want to upload an image/video so that I can quickly check if it has been manipulated.
* **Result**: Pass
* **Evidence**:
  1. User navigates to “Upload” page
  2. User selects valid .mp4 file and clicks Upload
  3. System analyzes file and shows result “AI-generated (87% confidence)”

**AC02 – Show suspicious areas (heatmap/highlighted frames)**

* **User Story**: As an investigator, I want to see visual indicators so that I can identify suspicious areas.
* **Result**: Pass
* **Evidence**:
  1. User uploads manipulated image
  2. System completes analysis
  3. Dashboard displays heatmap overlay with highlighted anomalies

**AC03 – Download PDF report**

* **User Story**: As an investigator, I want to download a PDF report for case files.
* **Result**: Pass
* **Evidence**:
  1. User uploads file and completes analysis
  2. User clicks “Export PDF”
  3. PDF downloaded with summary, heatmap snapshot, and confidence score

**AC04 – Metadata and anomaly scores**

* **User Story**: As a forensic analyst, I want to review metadata and anomaly scores for deeper verification.
* **Result**: Pass
* **Evidence**:
  1. User uploads test image
  2. Opens “Detailed View” tab
  3. System displays resolution, format, creation time + anomaly score list

**AC05 – Export JSON results**

* **User Story**: As a forensic analyst, I want to export results in JSON format for integration.
* **Result**: Pass
* **Evidence**:
  1. User uploads image
  2. Runs analysis
  3. Clicks “Export JSON” → JSON file downloads with metadata, anomaly scores, confidence values

**AC06 – Frame-by-frame video analysis**

* **User Story**: As a forensic analyst, I want to see frame-by-frame analysis for videos.
* **Result**: Pass
* **Evidence**:
  1. User uploads manipulated video
  2. Selects “Frame Analysis” tab
  3. System shows timeline with flagged suspicious frames

**AC07 – Report includes file hash + chain-of-custody**

* **User Story**: As a legal officer, I want reports to include file hashes and custody logs for court use.
* **Result**: Pass
* **Evidence**:
  1. User uploads video
  2. Generates PDF report
  3. Report includes SHA256 hash + chain-of-custody records

**AC08 – Offline functionality**

* **User Story**: As a client, I want the system to work offline in secure environments.
* **Result**: Pass
* **Evidence**:
  1. Internet disconnected
  2. User starts local application
  3. Uploads image → analysis runs successfully offline

**AC09 – Reliable confidence score**

* **User Story**: As a client, I want reliable confidence scores for every detection.
* **Result**: Pass
* **Evidence**:
  1. User uploads test image
  2. Analysis completes
  3. Result shows “Likely manipulated (92% confidence)”

**AC10 – Prioritize backlog by feedback**

* **User Story**: As a product owner, I want user feedback to prioritize backlog items.
* **Result**: Pass
* **Evidence**:
  1. User submits “Offline mode needed urgently” feedback
  2. Product backlog refreshes
  3. Story “Offline mode” moves to top priority

**AC11 – Modular detection components**

* **User Story**: As a developer, I want modular detection components so I can replace models.
* **Result**: Pass
* **Evidence**:
  1. Developer replaces detection model file
  2. System reloads module
  3. New model used successfully in analysis

**AC12 – Logs for debugging**

* **User Story**: As a developer, I want logs of every analysis for debugging/audit.
* **Result**: Pass
* **Evidence**:
  1. User uploads file
  2. Runs analysis
  3. Log file created with timestamp, filename, detection model, result

**AC13 – Simple verification for journalists/public**

* **User Story**: As a journalist/public user, I want to verify authenticity easily.
* **Result**: Pass
* **Evidence**:
  1. User pastes online video link
  2. Clicks “Verify”
  3. System displays “Likely authentic” with confidence percentage

**AC01 – Upload image/video for manipulation check**

* **Test Type**: Functional
* **Execution Type**: Manual
* **Objective**: Verify if a valid image/video can be uploaded and analyzed successfully.
* **Setup**:
  + System running in normal mode.
  + User logged in.
* **Pre-Conditions**:
  + User has a valid image or video file (≤2GB, supported format: .jpg, .png, .mp4).
* **Notes**:
  + Navigate to the upload page.
  + Select a valid test file (test\_video.mp4).
  + Click “Upload”.
  + Wait for system analysis to complete.
  + Verify that detection result is displayed with confidence score (e.g., “AI-generated: 87%”).
  + Verify result is saved in dashboard history.

**AC02 – Show suspicious areas (heatmap/highlighted frames)**

* **Test Type**: Functional
* **Execution Type**: Manual
* **Objective**: Verify that the system displays highlighted suspicious areas after analysis.
* **Setup**:
  + System running with visualization module enabled.
* **Pre-Conditions**:
  + User uploaded a manipulated file.
* **Notes**:
  + Upload a manipulated image containing edited regions.
  + Wait for analysis to finish.
  + Open visualization tab in dashboard.
  + Verify suspicious regions are clearly highlighted (heatmap overlay).
  + Confirm that explanation tooltips show anomaly scores when hovering.

**AC03 – Download PDF report**

* **Test Type**: Functional
* **Execution Type**: Manual
* **Objective**: Verify user can export a PDF report including detection results.
* **Setup**:
  + Reporting module active.
* **Pre-Conditions**:
  + A file has been analyzed successfully.
* **Notes**:
  + Upload an image and complete analysis.
  + Click “Export PDF” button.
  + System generates and downloads PDF.
  + Verify PDF contains:
    - File name and type
    - Detection result with confidence score
    - Suspicious area visualization snapshot
    - Timestamp of analysis

**AC04 – Metadata and anomaly scores**

* **Test Type**: Functional
* **Execution Type**: Manual
* **Objective**: Verify detailed metadata and anomaly scores are displayed.
* **Setup**:
  + Metadata extraction module active.
* **Pre-Conditions**:
  + Valid file uploaded.
* **Notes**:
  + Upload a test image.
  + Open “Detailed View” tab.
  + Verify metadata fields: filename, resolution, creation time, format.
  + Verify anomaly scores are listed with percentage values.
  + Confirm anomaly score matches visualization highlights.

**AC05 – Export JSON results**

* **Test Type**: Functional
* **Execution Type**: Manual
* **Objective**: Verify JSON export works with detection details.
* **Setup**:
  + JSON export module active.
* **Pre-Conditions**:
  + A file has been analyzed.
* **Notes**:
  + Upload a manipulated image.
  + Run analysis to completion.
  + Click “Export JSON”.
  + Verify JSON file is downloaded.
  + Open JSON and confirm it contains: metadata, anomaly scores, confidence score, suspicious regions.

**AC06 – Frame-by-frame video analysis**

* **Test Type**: Functional
* **Execution Type**: Manual
* **Objective**: Verify system provides flagged frames in video.
* **Setup**:
  + Video analysis module active.
* **Pre-Conditions**:
  + Valid video file uploaded.
* **Notes**:
  + Upload manipulated\_clip.mp4.
  + Wait until analysis completes.
  + Open “Frame Analysis” tab.
  + Verify suspicious frames are highlighted in timeline.
  + Confirm user can click a frame to preview anomaly.

**AC07 – Report includes file hash + chain-of-custody**

* **Test Type**: Functional
* **Execution Type**: Manual
* **Objective**: Verify generated report includes integrity data.
* **Setup**:
  + Hash calculation + custody module active.
* **Pre-Conditions**:
  + A PDF report is generated.
* **Notes**:
  + Upload and analyze test file.
  + Click “Export PDF”.
  + Open PDF report.
  + Verify it contains SHA256 hash.
  + Verify chain-of-custody log with timestamps.

**AC08 – Offline functionality**

* **Test Type**: Functional
* **Execution Type**: Manual
* **Objective**: Verify system functions without internet.
* **Setup**:
  + Application installed locally.
* **Pre-Conditions**:
  + Internet disconnected.
* **Notes**:
  + Start application offline.
  + Upload a valid image.
  + Verify analysis runs successfully.
  + Confirm result is displayed without requiring server connection.

**AC09 – Reliable confidence score**

* **Test Type**: Functional
* **Execution Type**: Manual
* **Objective**: Verify detection result always includes a confidence score.
* **Setup**:
  + Detection model active.
* **Pre-Conditions**:
  + Valid file uploaded.
* **Notes**:
  + Upload manipulated video.
  + Wait for analysis.
  + Verify result includes confidence score (e.g., 92%).
  + Confirm score is consistent with anomaly visualization.

**AC10 – Prioritize backlog by feedback**

* **Test Type**: Functional
* **Execution Type**: Manual
* **Objective**: Verify backlog reflects client feedback priority.
* **Setup**:
  + Feedback system active.
* **Pre-Conditions**:
  + Client submits feedback.
* **Notes**:
  + Enter feedback as “High priority: offline mode”.
  + Submit feedback.
  + Open backlog dashboard.
  + Verify story “Offline Mode” appears higher in priority list.

**AC11 – Modular detection components**

* **Test Type**: Functional
* **Execution Type**: Manual
* **Objective**: Verify new detection component can be swapped in without breaking system.
* **Setup**:
  + Modular framework active.
* **Pre-Conditions**:
  + New detection model available.
* **Notes**:
  + Replace existing detection model with new version file.
  + Restart application.
  + Upload test image.
  + Verify analysis runs successfully with new model.

**AC12 – Logs for debugging**

* **Test Type**: Functional
* **Execution Type**: Manual
* **Objective**: Verify logs are generated for every analysis.
* **Setup**:
  + Logging module active.
* **Pre-Conditions**:
  + File available for upload.
* **Notes**:
  + Upload sample.jpg.
  + Wait for analysis.
  + Open logs directory.
  + Verify log entry with timestamp, file name, model version, and confidence score.

**AC13 – Simple verification for journalists/public**

* **Test Type**: Functional
* **Execution Type**: Manual
* **Objective**: Verify system provides simple pass/fail verification for public users.
* **Setup**:
  + Public interface active.
* **Pre-Conditions**:
  + User has a video link.
* **Notes**:
  + Copy YouTube video link.
  + Paste into verification input.
  + Click “Verify”.
  + Verify system displays simple result: “Likely AI-generated” or “Likely Authentic”.