

# **Software Test Documentation**

## **Test Plan Identifier:**

TP-Online Proctoring

## **Introduction**

This test plan outlines the comprehensive testing process for the online proctoring system. It employs a structured approach, including Unit Testing, Integration Testing, System Testing, and Acceptance Testing, to ensure that the system meets the necessary standards for performance, functionality, and user acceptance.

## **Test Items**

- Face Detection Module
- Object Detection Module
- ID Verification Module
- Window Switching Monitoring Module
- Anti-Flagging System Module
- Audio Recording/Voice Recording Module
- User Interface (Login and Password) Module
- Webcam and Microphone Access Module
- Test Paper Testing (Mock and Exam Paper)
- APIs Testing
- Database Update Module

## **Characteristics of their transmittal media**

- All modules will be tested in a web-based environment with internet connectivity.

## **Features to be Tested**

- All functionalities of the individual modules as outlined in the unit testing section.
- Integration between modules as outlined in the integration testing section.
- Overall system performance and user acceptance as outlined in the system and acceptance testing sections.

## **Features Not to Be Tested**

None specified at this stage

## **Approach**

### **Overall Approach to Testing**

- The testing will follow a structured methodology, divided into four main categories: Unit Testing, Integration Testing, System Testing, and Acceptance Testing.

### **Testing Techniques and Tools**

- Manual testing for functional validation.
- Automated testing for regression and performance scenarios.

### **Item Pass/Fail Criteria**

- A module is considered passed if it meets the expected results as outlined in the test scenarios.

### **Suspension Criteria and Resumption Requirements**

- Testing will be suspended if critical issues arise that prevent further testing.
- When testing resumes, any affected tests will be redone to ensure accuracy.

### **Test Deliverables**

- Test Plan Document
- Test Design Specifications
- Test Case Specifications
- Test Procedure Specifications
- Test Summary Reports

# 1. Test Plan Document

**Purpose:** The Test Plan Document serves as a foundational guide that outlines the overall strategy for testing a software application. It provides a structured approach to ensure that all aspects of the software are tested effectively and that the testing process aligns with project objectives.

**Introduction:** This section offers a brief overview of the project, its objectives, and the importance of the testing process within the software development lifecycle.

**Scope of Testing:** This part defines what will and will not be included in the testing process, specifying the features, modules, and functionalities to be evaluated.

**Testing Strategy:** A high-level description of the testing types to be utilized, including unit testing, integration testing, system testing, and acceptance testing. It outlines the rationale behind choosing specific testing approaches based on project requirements.

**Test Environment:** Detailed information regarding the hardware and software configuration necessary for executing the tests, including specifications for servers, databases, and any third-party tools.

**Resources:** Identification of the testing team members, their roles, responsibilities, and qualifications to ensure that the right skills are applied during testing.

**Schedule:** A timeline for testing activities, including key milestones, deadlines, and any dependencies that may affect the testing schedule.

**Risk Management:** A proactive approach to identifying potential risks that could impact testing and outlining strategies for mitigation. This includes assessing both technical and project management risks.

**Approval and Sign-off:** Criteria for determining test completion, including the metrics for success and who will be responsible for approving the final test results.

## 2. Test Design Specifications

**Purpose:** The Test Design Specifications document serves to define the design aspects of test cases, and the methodologies used to derive them. It ensures that all testing is methodical and traceable to requirements.

**Introduction:** An overview of the purpose of the document, emphasizing its role in maintaining the quality of the testing process.

**Test Items:** A comprehensive list of the features and functionalities to be tested, organized by priority or relevance.

**Test Design Techniques:** Explanation of the methodologies used to design test cases, such as boundary value analysis, equivalence partitioning, state transition testing, and more. This section justifies the selection of techniques based on the complexity of the application.

**Test Data Requirements:** Specifics about the data needed for testing, including formats, sources, and any conditions for generating or retrieving test data.

**Test Environment Requirements:** Detailed information about the configuration needed to perform the tests effectively, including setup for both hardware and software.

**Traceability:** A matrix linking requirements to corresponding test cases, ensuring that all requirements are covered by specific tests.

### 3. Test Case Specifications

**Purpose:** Test Case Specifications provide detailed documentation of individual test cases, outlining the precise conditions under which each test will be executed.

**Test Case ID:** A unique identifier assigned to each test case for tracking purposes.

**Test Case Name:** A descriptive name summarizing the purpose of the test case.

**Objective:** A clear statement of the goal of the test case, specifying what functionality is being tested.

**Preconditions:** A description of the necessary conditions that must be fulfilled before executing the test, such as user authentication or data availability.

**Test Steps:** A sequential outline of the specific actions required to execute the test case, including any setup activities.

**Expected Results:** A detailed description of what the expected outcome should be if the system behaves correctly, providing a clear benchmark for comparison.

**Status:** An indication of whether the test case passed, failed, or was blocked, along with reasons for failure if applicable.

## 4. Test Procedure Specifications

**Purpose:** Test Procedure Specifications describe the specific procedures required to execute the tests, ensuring consistency and reproducibility throughout the testing process.

**Introduction:** Overview of the procedures to be followed during testing and their significance in achieving quality assurance.

**Test Procedure ID:** Unique identifiers for each test procedure to facilitate tracking.

**Test Procedure Name:** A title summarizing the specific procedure.

**Prerequisites:** Conditions that must be met before executing the test, such as specific configurations or setups.

**Test Execution Steps:** Detailed instructions for executing the tests, including preparatory actions, execution steps, and post-test activities.

**Roles and Responsibilities:** Assignment of specific tasks to team members to ensure accountability and clarity during the testing process.

**Environment Setup:** Clear instructions for configuring the test environment, ensuring all necessary components are in place for effective testing.

**Data Preparation:** Guidelines for preparing any required test data, including examples and formatting instructions.

## 5. Test Summary Reports

**Purpose:** Test Summary Reports provide a high-level overview of the testing activities, summarizing results and insights gathered during the testing process. These reports serve as a communication tool for stakeholders to understand the quality of the software.

**Introduction:** A brief overview of the report's objectives and its importance in the context of the project.

**Test Execution Summary:** A summary of the total number of tests executed, including the number of tests passed, failed, and blocked, along with associated metrics.

**Defect Summary:** An overview of defects identified during testing, categorized by severity, status, and any trends observed.

**Test Coverage:** Metrics indicating the extent to which the application has been tested, including coverage of requirements and functionalities.

**Test Environment Summary:** Description of the environments used for testing, including configurations and any issues encountered.

**Recommendations:** Suggestions for improvements based on testing outcomes, including potential areas for further testing or development.

**Conclusion:** Final remarks on the testing process, the overall quality of the software, and readiness for deployment.

## Testing Tasks

- Preparation of testing environments.
- Execution of test cases for each module.
- Reporting and documenting test results.

## Environmental Needs

- Physical characteristics include:
- A computer lab with stable internet access.
- Required software for testing (browsers, testing tools).
- Security requirements will be determined based on user data privacy.

## Responsibilities

- **Test Management:** QA Team
- **Module Development:** Development Team
- **Testing Execution:** QA Team
- **User Feedback Collection:** User Support Team

## Staffing and Training Needs

- **Staffing Needs:** Testers with experience in software testing.
- **Training Options:** Workshops on the online proctoring system functionalities.

## Schedule

### Test Milestones:

- Unit Testing
- Integration Testing
- System Testing
- Acceptance Testing

## Risks and Contingencies

- **High-Risk Assumptions:** System performance under load conditions.
- **Contingency Plans:** If critical performance issues are identified, additional resources will be allocated for optimization.



## Use Case Specifications

Use Case ID	Use Case Name	Description	Preconditions	Postconditions	Primary Actor	Trigger
UC-001	Face Detection Module	Verify accurate face detection functionality.	The system is online, and the camera is activated.	Faces are detected and outlined correctly.	Examinee	The exam starts, and the webcam is enabled.
UC-002	Object Detection Module	Detect unauthorized objects like mobile phones or books.	The camera is active and monitoring the exam area.	Prohibited items are flagged for violations.	Examinee	The exam starts with objects in view.
UC-003	ID Verification Module	Validate the examinee's identity against a photo ID.	The examinee has a pre-registered ID in the system.	The identity is successfully matched.	Examinee	The exam starts, and ID verification is initiated.
UC-004	Window Switching Monitoring Module	Detect any attempt to switch away from the exam window.	The exam is in progress.	Alerts are generated for any unauthorized actions.	Examinee	The examinee attempts to switch windows.
UC-005	Anti-Flagging System Module	Issue warnings before terminating the exam for violations.	The exam is in progress.	Warnings are issued based on violation counts.	Examinee	An unauthorized action is detected.
UC-006	Audio Recording/Voice Recording Module	Record audio during the exam.	The microphone is enabled at the exam start.	Audio is recorded accurately and logged.	Examinee	The exam starts, and the microphone is enabled.
UC-007	User Interface (Login and Password) Module	Validate secure login and access.	User must have valid credentials.	User is logged in or denied access.	Examinee	User attempts to log in.
UC-008	Webcam and Microphone Access Module	Verify webcam and microphone access during the exam.	The exam is initiated.	Both devices activate without errors.	Examinee	The exam starts.
UC-009	Test Paper Testing (Mock and Exam Paper)	Test uploading and accessing mock and exam papers.	The user is logged in.	Test papers are accessible in all formats.	Examinee	User uploads exam papers.
UC-010	APIs Testing	Validate proper functioning of APIs for data retrieval.	The system is functional and connected.	API calls return expected responses.	System	Various API calls are performed.
UC-011	Database Update Module	Ensure database consistency and updates.	The system is logging actions.	Database actions are logged correctly.	System	User performs actions that trigger updates.

## TESTING TRACKING SHEET

S.NO	MODULE TESTED/SCENARIO	INPUT	TESTING TYPE	OUTPUT
1.	Face Detection Module	Live webcam feed	Unit Testing	Accurate detection of facial features
2.	Object Detection Module	Real-time video stream	Unit Testing	Detection of prohibited objects in the video feed
3.	ID Verification Module	User-submitted ID image	Unit Testing	Successful verification of ID authenticity
4.	Anti-Flagging System Module	User's actions (e.g., multiple window switches)	Unit Testing	Trigger alerts for suspicious behaviour
5.	Audio Recording/Voice Recording Module	Audio input from microphone	Unit Testing	Clear recording of user's audio
6.	User Interface (Login and Password) Module	User login credentials	Unit Testing	Successful login with correct credentials
7.	Webcam and Microphone Access Module	Request for access to webcam and microphone	Unit Testing	Granting or denial of access
8.	Test Paper Testing (Mock and Exam Paper)	Sample exam paper input	Unit Testing	Successful loading and submission of the test paper
9.	APIs Testing	Various API requests for different functionalities	Unit Testing	Proper responses from APIs
10.	Database Update Module	Inputs for student details, test results, etc.	Unit Testing	Successful updates in the database
11.	Window Switching Monitoring Module	User switching between different windows	Unit Testing	Detection and logging of window switching activity
12.	Face Detection + ID Verification Integration	Webcam feed + ID verification request	Integration Testing	Combined results for face match and ID authenticity
13.	Object Detection + Anti-Flagging System Integration	Real-time video stream with potential prohibited objects	Integration Testing	Detection and alert generation for prohibited objects
14.	Window Switching Monitoring + Anti-Flagging System Integration	User's window switching actions	Integration Testing	Alerts triggered for excessive switching
15.	Audio Recording + Webcam Access Integration	User's audio and webcam access requests	Integration Testing	Simultaneous recording and video feed
16.	APIs + Database Update Integration	API requests involving data updates	Integration Testing	Synchronization between API responses and database updates
17.	End-to-End Exam Simulation	Full exam scenario with multiple modules	System Testing	Successful completion of exam with monitoring and recording

18.	<b>Multiple User Load Testing</b>	<b>Simultaneous login and exam-taking by multiple users</b>	<b>System Testing</b>	<b>System performance under load</b>
19.	<b>Low Bandwidth Conditions</b>	<b>Exam simulation under low network bandwidth</b>	<b>System Testing</b>	<b>Graceful handling of audio/video lag or interruptions</b>
20.	<b>Long Duration Testing</b>	<b>Extended exam session</b>	<b>System Testing</b>	<b>System stability over long duration</b>
21.	<b>User Acceptance Testing</b>	<b>User feedback during mock exam</b>	<b>Acceptance Testing</b>	<b>Feedback on ease of use and issues encountered</b>
22.	<b>Proctor Review</b>	<b>Manual review by a proctor of recorded sessions</b>	<b>Acceptance Testing</b>	<b>Identification of any suspicious activities</b>
23.	<b>Login Process Feedback</b>	<b>User feedback on login experience</b>	<b>Acceptance Testing</b>	<b>Usability improvements based on feedback</b>