# DYSLEXILENS

AAYUSH KUMAR SHRIVASTAVA

ABHISHEK VERMA

ARCHIT GOEL

KARTIK VERMA

**TABLE**

# 1.Introduction

This document presents a comprehensive framework for testing the Early Detection of Dyslexia system, focusing on ensuring its effectiveness in identifying early signs of dyslexia through handwriting analysis. It outlines structured testing strategies and methodologies tailored specifically to evaluate the system's ability to detect dyslexia accurately and reliably.

The plan emphasizes validating both functional and non-functional requirements, ensuring the system performs robustly under various conditions. Testing phases—such as unit, integration, system, and acceptance testing—are meticulously designed to verify each component's functionality. For instance, unit testing will assess individual modules, like Optical Character Recognition (OCR) for handwriting conversion and machine learning algorithms for pattern analysis. Integration testing ensures seamless communication between modules, such as text extraction and dyslexia prediction analysis.

System testing evaluates the complete workflow, simulating real-world scenarios to confirm that the system can consistently detect dyslexic patterns, such as irregular letter spacing, slanting, and coherence. Acceptance testing focuses on validating the system against user requirements, ensuring it meets the needs of educators and healthcare professionals.

Resource allocation includes specialized tools and environments to mimic educational and clinical settings, providing realistic data for testing. By following these strategies, the document ensures that the system is accurate, user-friendly, and scalable, enhancing its potential to support early dyslexia intervention efforts effectively.

## 1.1 Scope

### 1.1.1 In Scope

The following functionalities and components will be tested:

- Handwriting recognition accuracy (OCR).
- Machine learning models (Decision Trees and Random Forests).
- Cognitive quiz assessment module.
- System performance under various loads.
- User interface and user experience (UI/UX).

### 1.1.2 Out of Scope

The following aspects will not be tested:

- Third-party OCR engine accuracy.

- Integration with external hardware or mobile devices.
- Real-time handwriting input (only pre-scanned images are evaluated).

## 1.2 Quality Objective

- Ensuring the system accurately detects dyslexia indicators.
- Validating all functional and non-functional requirements.
- Identifying and resolving critical bugs before deployment.
- Delivering a user-friendly, reliable, and scalable solution.

## 1.3 Roles and Responsibilities

- Test Manager – Dr. Kalpana Sagar
- Developers – Aayush Kumar Shrivastava , Kartik Verma
- ML algo Developers – Abhishek Verma , Archit Goel
- Installation Team - Aayush Kumar Shrivastava , Kartik Verma , Abhishek Verma , Archit Goel

# 2.Test Methodology

## 2.1Overview

The **Agile methodology** is chosen for its flexibility, iterative approach, and ability to handle changing requirements efficiently.

## 2.2Test Levels

**1.Unit Testing**

Unit testing focuses on validating individual components or modules of the system in isolation. In the context of a dyslexia detection model, this involves testing key functionalities such as Optical Character Recognition (OCR) for accurate text extraction, and machine learning (ML) algorithms like decision trees or random forests for proper data processing and classification. Each unit is tested independently to ensure that it performs as expected, with specific attention to edge cases and error handling. This phase helps identify bugs early in development, ensuring each component meets its design specifications.

**2. Integration Testing:**

Integration testing examines how different modules of the system work together. For a dyslexia detection tool, this involves validating the seamless communication between components such as text extraction from images (OCR) and subsequent analysis by the machine learning model. The goal is to ensure that data flows correctly between modules and that interfaces between different software components function smoothly. This step often uncovers issues related to data formats, synchronization, and communication protocols that may not be apparent during unit testing.

**3. System Testing:**

System testing evaluates the entire systems functionality from end-to-end. It simulates real-world usage scenarios to verify that the system as a whole meets the specified requirements. For instance, in a dyslexia detection application, system testing would involve processing sample handwriting inputs through the complete pipeline from image upload to text extraction, analysis by the ML model, and the generation of diagnostic results. This phase ensures that all components work together harmoniously, and that the system performs correctly under various conditions, such as different handwriting styles or text lengths.

**4. Acceptance Testing:**

Acceptance testing involves validating the system against user requirements and ensuring it meets the needs of its intended users. This phase typically involves real-world users, such as teachers or healthcare professionals, who test the system under actual operating conditions. The objective is to determine if the system delivers the expected outcomes and provides a user-friendly experience. For a dyslexia detection tool, acceptance testing might involve evaluating the systems accuracy, ease of use, and reliability in identifying dyslexic tendencies in various handwriting samples. This phase is crucial for ensuring that the system meets its functional and non-functional requirements and is ready for deployment.

## 2.3Test Completeness

- 100% test coverage
- All Manual & Automated Test cases executed
- All open bugs are fixed or will be fixed in next release

# 3.Test Deliverables

- **Test Plan Document**
- **Test Cases**
- **Requirement Traceability Matrix (RTM)**
- **Bug Reports**

# TEST CASES

## Boundary Value Analysis (BVA) for Dyslexia Detection System:

| | A | B | C |
|---|---|---|---|
| 1 | **Input Parameter** | **Boundary Value** | **Test Case Description** |
| 2 | **Handwriting Sample Length** | 1 character | Verify the system can handle minimal input without errors. |
| 3 | | 2–3 characters | Ensure the system processes very short but valid inputs correctly. |
| 4 | | 20–100 characters | Test typical input length for accurate analysis. |
| 5 | | 500–1000 characters | Assess performance with longer handwriting samples. |
| 6 | | 1000+ characters | Confirm system stability with large inputs. |
| 7 | **Spelling Accuracy Score** | 0% | Check if the system identifies extreme spelling errors as high dyslexia risk. |
| 8 | | 10%–20% | Validate detection of poor spelling accuracy indicating potential dyslexia. |
| 9 | | 60%–80% | Test average spelling performance classification. |
| 10 | | 90%–95% | Ensure high accuracy does not result in false positives. |
| 11 | | 100% | Confirm perfect spelling does not flag dyslexia erroneously. |
| 12 | **Quiz-Based Evaluation Score** | 0/10 | Verify high dyslexia risk classification with no correct answers. |
| 13 | | 2–3/10 | Assess detection accuracy for low performance. |
| 14 | | 5–7/10 | Check classification for medium quiz performance. |
| 15 | | 8–9/10 | Ensure correct evaluation for high quiz performance. |
| 16 | | 10-Oct | Validate no false negatives with perfect quiz scores. |
| 17 | | | |

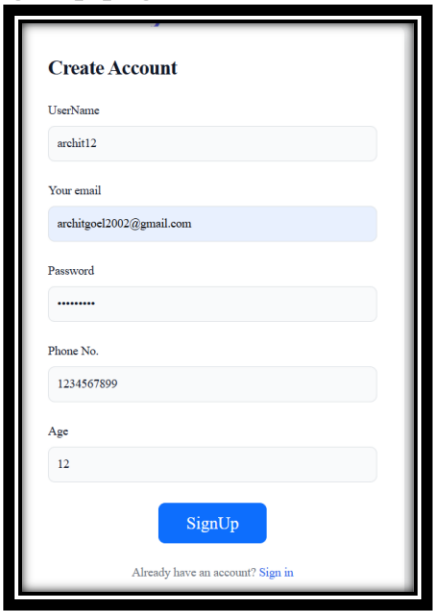
## Decision Tree Boundary Tests:

| | A | B | C | D |
|---|---|---|---|---|
| 1 | **Decision Node** | **Condition** | **True Branch** | **False Branch** |
| 2 | **Root Node: Spelling Accuracy** | Spelling Accuracy < 60% | High Dyslexia Risk | Move to Grammatical Accuracy Node |
| 3 | **Node 2: Grammatical Accuracy** | Grammatical Accuracy < 70% | Medium Dyslexia Risk | Move to Phonetic Accuracy Node |
| 4 | **Node 3: Phonetic Accuracy** | Phonetic Errors > 3 | High Dyslexia Risk | Low Dyslexia Risk |
| 5 | **Leaf Node: Final Classification** | - | High, Medium, or Low Dyslexia Risk output based on prior conditions | - |
| 6 | | | | |

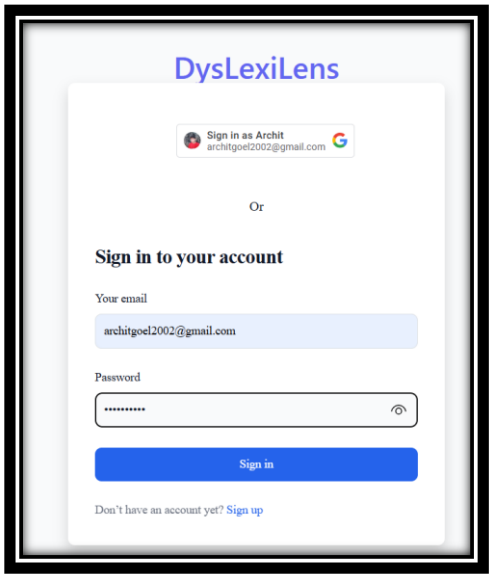
# SIGN UP AND LOGIN PAGE

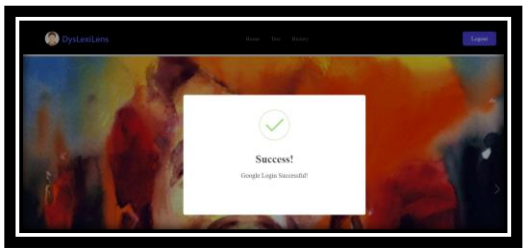
| SIGN UP PAGE | | | | | |
|---|---|---|---|---|---|
| serial Number | test Id | Input | Actual Output | Expected Output | Result |
| 1 | TC1 | Signup Credentials are correct | Signup Success | Signup Success | Pass |
| 2 | TC2 | Login Credentials are correct | Login Success | Login Success | Pass |
| 3 | TC3 | Signup Credentials are not correct | Signup Fail | Signup Fail | Pass |
| 4 | TC4 | Login Credentials are not correct | Login Fail | Login Fail | Pass |
| 5 | TC5 | Signup, UserName Not Provided | Signup Fail | Signup Fail | Pass |
| 6 | TC6 | Login, UserName or Password wrong | Login Fail | Login Fail | Pass |

1. Sign up page

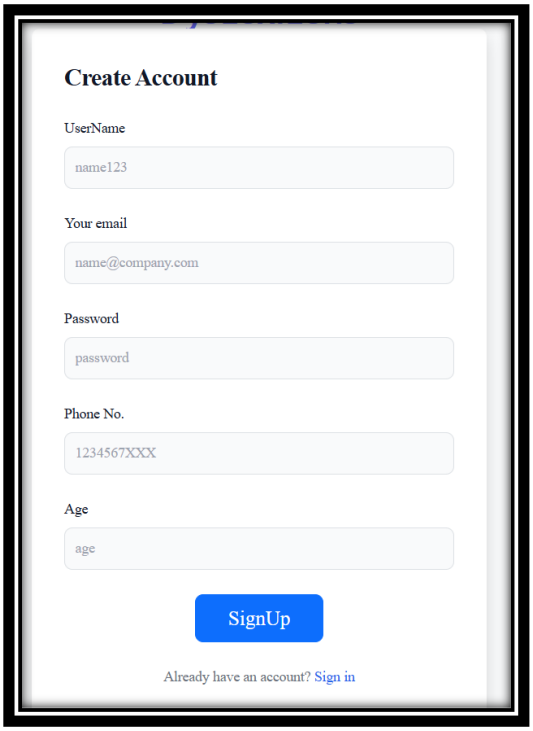


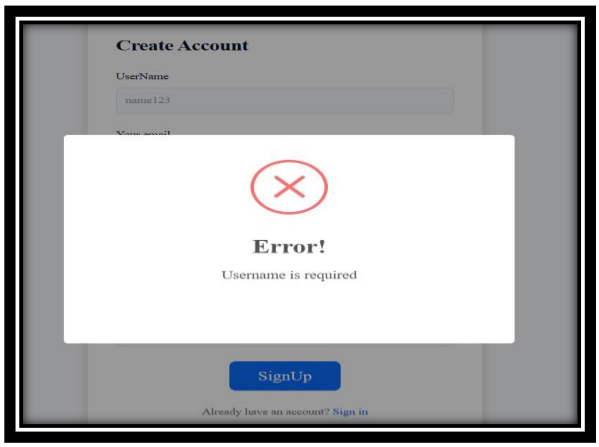

2. Login page




3. Sign up page username not written



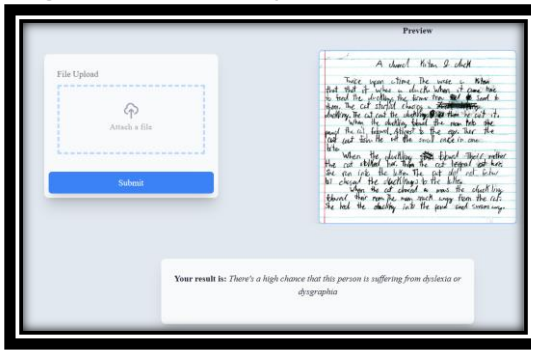

4. Password not valid

# MODEL TEST CASES

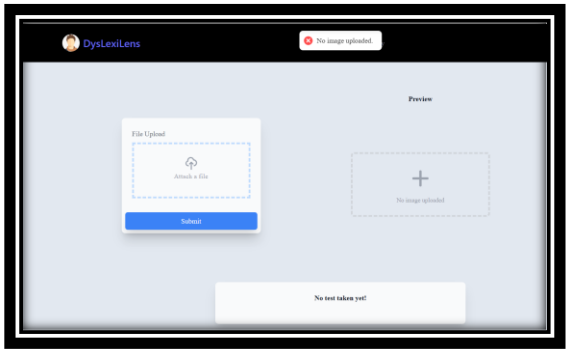
| MODEL | | | | | |
|---|---|---|---|---|---|
| serial Number | test Id | Input | Actual Output | Expected Output | Result |
| 1 | TC1 | If Image uploaded with phone camera | Dyslexia Predicted | Non Dyslexic | Fail |
| 2 | TC2 | If Test Dyslexic Image is uploaded in model | Dyslexia Predicted | Dyslexia Predicted | Pass |
| 3 | TC3 | If Test Non Dyslexic Image is uploaded | Non Dyslexic Predicted | Non Dyslexic Predicted | Pass |
| 4 | TC4 | If Quiz attempted by non Dyslexic | Non Dyslexic Predicted | Non Dyslexic Predicted | Pass |
| 5 | TC5 | If Wrong Image is upoladed | Dylexic Predicted | Non Identifiable | Fail |
| 6 | TC6 | If Photo is uploaded | Non Identifiable | Non Identifiable | Pass |

- Slim chances of dyslexia

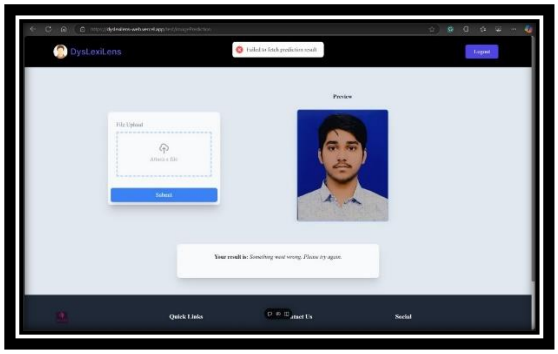

- High chances of dyslexia



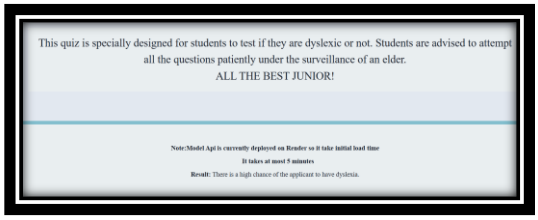

- Handwritten image not provided




- If Photo is uploaded



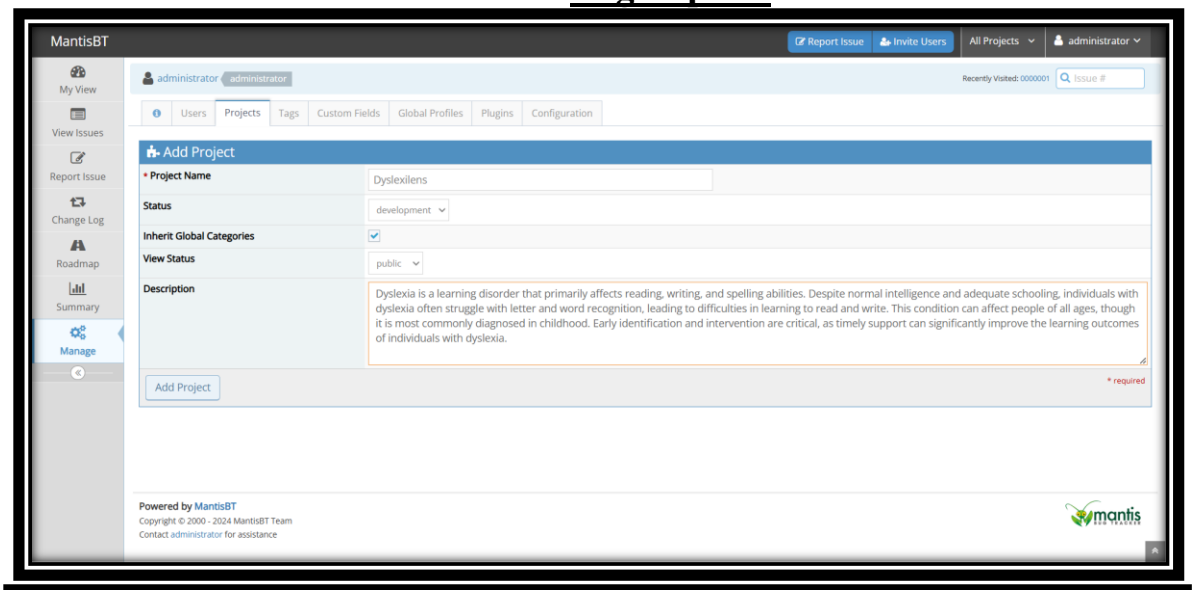

- Quiz module attempted all questions

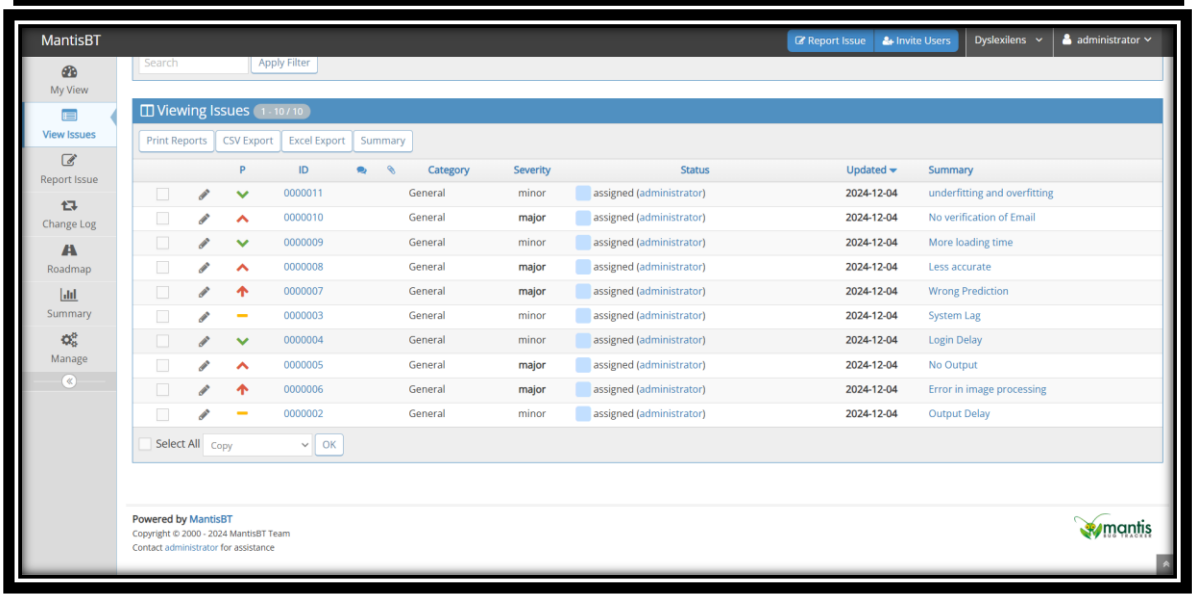


# Requirement Traceability Matrix (RTM)

| Serial Number | Test Id | Input | Actual Output | Expected Output | Result |
|---|---|---|---|---|---|
| 1 | TC1 | Signup Credentials | Signup Success | Signup Success | Pass |
| 2 | TC2 | Login Credentials | Login Success | Login Success | Pass |
| 3 | TC3 | Signup, UserName Not Provided | Signup Fail | Signup Fail | Pass |
| 4 | TC4 | Login, UserName or Password wrong | Login Fail | Login Fail | Pass |
| 5 | TC5 | If Image uploaded with phone camera | Dyslexia Predicted | Non Dyslexic | Fail |
| 6 | TC6 | If Test Dyslexic Image is uploaded in model | Dyslexia Predicted | Dyslexia Predicted | Pass |
| 7 | TC7 | If Test Non Dyslexic Image is uploaded | Non Dyslexic Predicted | Non Dyslexic Predicted | Pass |
| 8 | TC8 | If Quiz attempted by non Dyslexic | Non Dyslexic Predicted | Non Dyslexic Predicted | Pass |
| 9 | TC9 | If Wrong Image is upoladed | Dylexic Predicted | Non Identifiable | Fail |
| 10 | TC10 | If Photo is uploaded | Non Identifiable | Non Identifiable | Pass |

# Bug Reports



| ID | Project | Reporter | Assigned To | Priority | Severity | Reproducibility | Product Version | Category | Date Submitted | OS | OS Version | Platform | View Status | Updated | Summary | Status | Resolution | Fixed in Version |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | Dyslexilens | administrator | administrator | normal | minor | sometimes | | General | 2024-12-04 | | | | public | 2024-12-04 | Output Delay | assigned | open | |
| 3 | Dyslexilens | administrator | administrator | normal | minor | sometimes | | General | 2024-12-04 | | | | public | 2024-12-04 | System Lag | assigned | open | |
| 4 | Dyslexilens | administrator | administrator | low | minor | sometimes | | General | 2024-12-04 | | | | public | 2024-12-04 | Login Delay | assigned | open | |
| 5 | Dyslexilens | administrator | administrator | high | major | sometimes | | General | 2024-12-04 | | | | public | 2024-12-04 | No Output | assigned | open | |
| 6 | Dyslexilens | administrator | administrator | urgent | major | sometimes | | General | 2024-12-04 | | | | public | 2024-12-04 | Error in image processing | assigned | open | |
| 7 | Dyslexilens | administrator | administrator | urgent | major | sometimes | | General | 2024-12-04 | | | | public | 2024-12-04 | Wrong Prediction | assigned | open | |
| 8 | Dyslexilens | administrator | administrator | high | major | always | | General | 2024-12-04 | | | | public | 2024-12-04 | Less accurate | assigned | open | |
| 9 | Dyslexilens | administrator | administrator | low | minor | sometimes | | General | 2024-12-04 | | | | public | 2024-12-04 | More loading time | assigned | open | |
| 10 | Dyslexilens | administrator | administrator | high | major | always | | General | 2024-12-04 | | | | public | 2024-12-04 | No verification of Email | assigned | open | |
| 11 | Dyslexilens | administrator | administrator | low | minor | sometimes | | General | 2024-12-04 | | | | public | 2024-12-04 | underfitting and overfitting | assigned | open | |

# 4.Resource & Environment Needs

## 4.1Testing Tools

**Bug Tracking Tool:** MantisBT
Mantis Bug Tracker (MantisBT) is an open-source, web-based bug tracking system. It provides project managers and developers with an organized way to manage software defects and improvements.

**Why Use MantisBT?**

• Cost-Effective: Open-source and free.
• Scalable: Suitable for small to large projects.
• Community Support: Strong community contributions ensure frequent updates and new features.
• Easy Deployment: Quick setup and simple to host.

## 4.2Test Environment

- **Operating System:** Windows 10 and above
- **Database:** MySQL
- **Web Server:** Apache
- **Development Frameworks:** Python, Scikit-Learn, TensorFlow
- **Additional Software:** Microsoft Office 2013+, Google Chrome, Visual Studio Code

# 5.Terms/Acronyms

| TERM/ACRONYM | DEFINITION |
|---|---|
| API | Application Program Interface |

| TERM/ACRONYM | DEFINITION |
|---|---|
| AUT | Application Under Test |
| OCR | Optical Character Recognition |
| RTM | Requirement Traceability Matrix |