

# EchoText

## Test Case Specification Version 1.01

Team: Ahmad Ghaddar, Kevin Xing, Harsh Bhagat, Andy Huang

## Revision History

Date	Description	Author	Comments
11/15/24	Version 1.00	Andy Huang	First Revision
12/3/24	Version 1.01	Andy Huang	Minor changes in each tables on each section.

## Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

Signature	Printed Name	Title	Date
	Kevin Xing	Team Lead Meeting Minutes	
	Ahmad Ghaddar	Presentation Lead Frontend Lead UI/UX Lead	
	Harsh Bhagat	GitHub Lead Backend and Data Management Lead	
	Andy Haung	Documentation Lead Testing/QA Lead	

## Table of Contents

<b>REVISION HISTORY .....</b>	<b>1</b>
<b>DOCUMENT APPROVAL .....</b>	<b>1</b>
<b>1. UNIT TESTING.....</b>	<b>1</b>
1.1 Database Class Unit Test Items .....	1
1.2 TextParser Class Unit Test Items.....	28
1.3 PasswordManager Class Unit Test Items.....	35
1.4 CombineAudioFiles Function Unit Test Items .....	42
1.5 TtsLogic Class Unit Test Items .....	44
1.6 FileDirectory Class Unit Test Items .....	53
1.7 VoiceProfileManager Class Unit Test Items .....	58
1.8 Basic UI Unit Test Items .....	66
<b>2. INTEGRATION TESTING.....</b>	<b>72</b>
2.1 Text-to-Speech Conversion .....	72
2.2 Document Conversion .....	73
2.3 Media Control.....	74
2.4 Voice Profile Creation .....	76
2.5 Passcode Creation .....	77
2.6 Passcode Validation .....	78
2.7 Recovery Code Functionality.....	79
2.8 Passcode Removal.....	80
2.9 Audio Search Functionality.....	81
2.10 Document Search Functionality.....	82
2.11 Document Import Process.....	83
2.12 File Rename and Delete .....	84
2.13 Batch File Deletion.....	85
2.14 File Sharing Functionality .....	86
2.15 Progress Indication System .....	87
<b>3. SYSTEM TESTING .....</b>	<b>89</b>
3.1 Core Application Functionalities.....	89
3.2 File and Queue Managment.....	92
3.3 User Settings and Customization .....	96

# 1. Unit Testing

## 1.1 Database Class Unit Test Items

ID	UT01
Title	Database Initialization and Connection
Test Description	Verify the database is initialized and connected properly.
Pre-conditions	<ul style="list-style-type: none"><li>- The application environment is properly set up.</li><li>- No existing database file at the expected location.</li></ul>
Test Steps	<ol style="list-style-type: none"><li>1. Initialize the Database class instance.</li><li>2. Invoke the openDatabase() method.</li><li>3. Observe the console output for any error messages.</li><li>4. Check the file system for the creation of the database file at the specified path.</li></ol>
Expected Results	<ol style="list-style-type: none"><li>1. Database file is created successfully if it doesn't exist.</li><li>2. Database connection is opened without errors.</li><li>3. Console prints "Database path: [path]" indicating successful connection.</li><li>4. No error messages are displayed.</li></ol>
Priority	High
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"><li>- The database file is created successfully.</li><li>- Database connection is opened without errors.</li><li>- Console displays the database path.</li><li>- No error messages are displayed.</li></ul> <p>Fail if:</p> <ul style="list-style-type: none"><li>- Database file is not created.</li><li>- Errors occur when opening the database connection.</li><li>- Console does not display the database path.</li><li>- Error messages are displayed.</li></ul>

ID	UT02
Title	Database Table Creation
Test Description	Verify that required database tables are created successfully.
Pre-conditions	<ul style="list-style-type: none"><li>- Database connection is open.</li><li>- No existing tables in the database.</li></ul>
Test Steps	<ol style="list-style-type: none"><li>1. Call the createTables() method.</li><li>2. Check the database schema for the Audios table.</li><li>3. Check for the ModelProfiles table.</li><li>4. Check for the Documents table.</li><li>5. Observe console output for any messages or errors.</li></ol>
Expected Results	<ul style="list-style-type: none"><li>- All required tables (Audios, ModelProfiles, Documents) are created successfully.</li><li>- Console outputs "Table created." for each table.</li><li>- No error messages are displayed.</li></ul>
Priority	High
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"><li>- All required tables are created.</li><li>- Console outputs "Table created." for each table.</li><li>- No error messages.</li></ul> <p>Fail if:</p> <ul style="list-style-type: none"><li>- Any table is not created.</li><li>- Errors occur during table creation.</li><li>- Console does not confirm creation.</li><li>- Error messages are displayed.</li></ul>

ID	UT03
Title	Database Deletion
Test Description	Verify that the database file is deleted successfully.
Pre-conditions	<ul style="list-style-type: none"><li>- Database file exists at the expected location.</li><li>- Database connection is closed.</li></ul>
Test Steps	<ol style="list-style-type: none"><li>1. Invoke the deleteDatabase() method.</li><li>2. Observe console output for confirmation or errors.</li><li>3. Check the file system to confirm the database file has been deleted.</li></ol>
Expected Results	<ul style="list-style-type: none"><li>- Database file is deleted successfully.</li><li>- Console prints "Database deleted successfully."</li><li>- No error messages are displayed.</li></ul>
Priority	Medium
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"><li>- Database file is deleted successfully.</li><li>- Console prints "Database deleted successfully."</li><li>- No error messages.</li></ul> <p>Fail if:</p> <ul style="list-style-type: none"><li>- Database file is not deleted.</li><li>- Errors occur during deletion.</li><li>- Console does not confirm deletion.</li><li>- Error messages are displayed.</li></ul>

ID	UT04
Title	Database Deletion with No Existing File
Test Description	Verify that the absence of a database file is handled gracefully.
Pre-conditions	<ul style="list-style-type: none"><li>- No database file exists at the expected location.</li><li>- Database connection is closed or not initialized.</li></ul>
Test Steps	<ol style="list-style-type: none"><li>1. Invoke the deleteDatabase() method.</li><li>2. Observe console output for error handling.</li><li>3. Ensure no exceptions are thrown during execution.</li></ol>
Expected Results	<ul style="list-style-type: none"><li>- Method handles absence of the database file gracefully.</li><li>- Console prints "Error deleting database: ..." with an appropriate message.</li><li>- No unhandled exceptions occur.</li></ul>
Priority	Low
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"><li>- Method handles absence gracefully.</li><li>- Console outputs an appropriate error message.</li><li>- No unhandled exceptions occur.</li></ul> <p>Fail if:</p> <ul style="list-style-type: none"><li>- Unhandled exceptions are thrown.</li><li>- Console does not display error messages.</li><li>- Method does not handle the missing file.</li></ul>

ID	UT05
Title	Insert Operations - Audios
Test Description	Verify that audio records are inserted accurately into the Audios table.
Pre-conditions	<ul style="list-style-type: none"><li>- Database is initialized and tables are created.</li><li>- Database connection is open.</li></ul>
Test Steps	<ol style="list-style-type: none"><li>1. Prepare test data for an audio record:<ul style="list-style-type: none"><li>- Name: "Test Audio"</li><li>- FilePath: "/path/to/test_audio.mp3"</li><li>- DateGenerated: "2024-11-18"</li><li>- Model: "TestModel"</li><li>- Pitch: 1.0</li><li>- Speed: 1.0</li><li>- DocumentId: nil</li><li>- TextFilePath: "/path/to/text_file.txt".</li></ul></li><li>2. Invoke insertAudio() with the test data.</li><li>3. Fetch all audio records using fetchAllAudios().</li><li>4. Verify that the new record exists with all fields accurately stored.</li></ol>
Expected Results	<ul style="list-style-type: none"><li>- Audio record is inserted into the Audios table.</li><li>- All fields are saved correctly in the database.</li><li>- Fetching records returns the inserted audio with accurate data.</li><li>- Console prints "Inserted audio with length: X seconds."</li></ul>
Priority	High
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"><li>- Audio record is inserted.</li><li>- All fields are stored correctly.</li><li>- Fetching records returns accurate data.</li><li>- Console outputs the insertion message.</li></ul> <p>Fail if:</p> <ul style="list-style-type: none"><li>- Record is not inserted.</li><li>- Fields are incorrect or missing.</li><li>- Error messages are displayed.</li></ul>



ID	UT06
Title	Insert Operations - Documents
Test Description	Verify that document records are inserted accurately into the Documents table.
Pre-conditions	<ul style="list-style-type: none"><li>- Database is initialized and tables are created.</li><li>- Database connection is open.</li></ul>
Test Steps	<ol style="list-style-type: none"><li>1. Prepare test data for a document record:<ul style="list-style-type: none"><li>- DocumentName: "Test Document"</li><li>- UploadDate: "2024-11-18"</li><li>- FilePath: "/path/to/test_document.pdf"</li><li>- FileType: "pdf".</li></ul></li><li>2. Invoke insertDocument() with the test data.</li><li>3. Fetch all document records using fetchAllDocuments().</li><li>4. Verify that the new record exists with all fields accurately stored.</li></ol>
Expected Results	<ul style="list-style-type: none"><li>- Document record is inserted into the Documents table.</li><li>- All fields are saved correctly in the database.</li><li>- Fetching records returns the inserted document with accurate data.</li><li>- No error messages are displayed.</li></ul>
Priority	High
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"><li>- Document record is inserted.</li><li>- All fields are stored correctly.</li><li>- Fetching records returns accurate data.</li><li>- No error messages are displayed.</li></ul> <p>Fail if:</p> <ul style="list-style-type: none"><li>- Record is not inserted.</li><li>- Fields are incorrect or missing.</li><li>- Error messages are displayed.</li></ul>

ID	UT07
Title	Insert Operations - ModelProfiles
Test Description	Verify that profile records are inserted accurately into the ModelProfiles table.
Pre-conditions	<ul style="list-style-type: none"><li>- Database is initialized and tables are created.</li><li>- Database connection is open.</li></ul>
Test Steps	<ol style="list-style-type: none"><li>1. Prepare test data for a profile:<ul style="list-style-type: none"><li>- ProfileName: "Test Profile"</li><li>- Pitch: 1.0</li><li>- Speed: 1.0</li><li>- Model: "TestModel".</li></ul></li><li>2. Invoke saveProfile() with the test data.</li><li>3. Fetch all profiles using fetchAllProfiles().</li><li>4. Verify that the new profile exists with all fields accurately stored.</li></ol>
Expected Results	<ul style="list-style-type: none"><li>- Profile is inserted into the ModelProfiles table.</li><li>- All fields are saved correctly in the database.</li><li>- Fetching profiles returns the inserted profile with accurate data.</li><li>- Console prints "Successfully inserted profile with id X."</li></ul>
Priority	High
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"><li>- Profile record is inserted.</li><li>- All fields are stored correctly.</li><li>- Fetching records returns accurate data.</li><li>- Console outputs the insertion message.</li></ul> <p>Fail if:</p> <ul style="list-style-type: none"><li>- Record is not inserted.</li><li>- Fields are incorrect or missing.</li><li>- Error messages are displayed.</li></ul>

ID	UT08
Title	Update Operations - Audios
Test Description	Verify that audio records are updated correctly in the Audios table.
Pre-conditions	<ul style="list-style-type: none"><li>- An existing audio record in the database.</li><li>- Database connection is open.</li></ul>
Test Steps	<ol style="list-style-type: none"><li>1. Fetch an existing audio record's ID.</li><li>2. Prepare updated data for one field (e.g., change Name to "Updated Audio").</li><li>3. Invoke updateAudio() with the updated data and existing ID.</li><li>4. Fetch the audio record again using fetchAllAudios().</li><li>5. Verify that only the updated field has changed and others remain the same.</li></ol>
Expected Results	<ul style="list-style-type: none"><li>- Audio record is updated successfully.</li><li>- The specified field is modified correctly.</li><li>- Other fields remain unchanged.</li><li>- Console prints "Successfully updated row."</li></ul>
Priority	Medium
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"><li>- Audio record is updated successfully.</li><li>- Only the specified field is modified.</li><li>- Other fields remain unchanged.</li><li>- Console outputs the update confirmation.</li></ul> <p>Fail if:</p> <ul style="list-style-type: none"><li>- Record is not updated.</li><li>- Incorrect fields are modified.</li><li>- Data integrity is compromised.</li><li>- Error messages are displayed.</li></ul>

ID	UT09
Title	Update Operations - Documents
Test Description	Verify that document records are updated correctly in the Documents table.
Pre-conditions	<ul style="list-style-type: none"><li>- An existing document record in the database.</li><li>- Database connection is open.</li></ul>
Test Steps	<ol style="list-style-type: none"><li>1. Fetch an existing document record's ID.</li><li>2. Prepare updated data for one field (e.g., change DocumentName to "Updated Document").</li><li>3. Invoke updateDocument() with the updated data and existing ID.</li><li>4. Fetch the document record again using fetchAllDocuments().</li><li>5. Verify that only the updated field has changed and others remain the same.</li></ol>
Expected Results	<ul style="list-style-type: none"><li>- Document record is updated successfully.</li><li>- The specified field is modified correctly.</li><li>- Other fields remain unchanged.</li><li>- Console prints "Successfully updated document in database."</li></ul>
Priority	Medium
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"><li>- Document record is updated successfully.</li><li>- Only the specified field is modified.</li><li>- Other fields remain unchanged.</li><li>- Console outputs the update confirmation.</li></ul> <p>Fail if:</p> <ul style="list-style-type: none"><li>- Record is not updated.</li><li>- Incorrect fields are modified.</li><li>- Data integrity is compromised.</li><li>- Error messages are displayed.</li></ul>

ID	UT10
Title	Update Operations - ModelProfiles
Test Description	Verify that profile records are updated correctly in the ModelProfiles table.
Pre-conditions	<ul style="list-style-type: none"><li>- An existing profile in the database.</li><li>- Database connection is open.</li></ul>
Test Steps	<ol style="list-style-type: none"><li>1. Fetch an existing profile's ID.</li><li>2. Prepare updated data (e.g., change Pitch to 1.5).</li><li>3. Invoke updateProfile() with the updated data.</li><li>4. Fetch the profile again using fetchAllProfiles().</li><li>5. Verify that the fields are updated accurately.</li></ol>
Expected Results	<ul style="list-style-type: none"><li>- Profile is updated successfully.</li><li>- Updated fields reflect new values.</li><li>- Console prints "Successfully updated profile with id X."</li></ul>
Priority	Medium
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"><li>- Profile is updated successfully.</li><li>- Updated fields reflect new values.</li><li>- Console outputs the update confirmation.</li></ul> <p>Fail if:</p> <ul style="list-style-type: none"><li>- Profile is not updated.</li><li>- Incorrect fields are modified.</li><li>- Data integrity is compromised.</li><li>- Error messages are displayed.</li></ul>

ID	UT11
Title	Fetch Operations - Audios
Test Description	Verify that all audio records are fetched correctly from the database.
Pre-conditions	<ul style="list-style-type: none"><li>- Database contains at least one audio record.</li><li>- Database connection is open.</li></ul>
Test Steps	<ol style="list-style-type: none"><li>1. Invoke fetchAllAudios() method.</li><li>2. Verify the data structure returned.</li><li>3. Check that all audio records are returned with correct data.</li></ol>
Expected Results	<ul style="list-style-type: none"><li>- All audio records are fetched successfully.</li><li>- Data returned matches the data in the database.</li><li>- No error messages are displayed.</li></ul>
Priority	High
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"><li>- All audio records are fetched successfully.</li><li>- Data returned matches the database.</li><li>- No error messages.</li></ul> <p>Fail if:</p> <ul style="list-style-type: none"><li>- Records are not fetched.</li><li>- Data is incorrect or incomplete.</li><li>- Error messages are displayed.</li></ul>

ID	UT12
Title	Fetch Operations - Documents
Test Description	Verify that all document records are fetched correctly from the database.
Pre-conditions	<ul style="list-style-type: none"><li>- Database contains at least one document record.</li><li>- Database connection is open.</li></ul>
Test Steps	<ol style="list-style-type: none"><li>1. Invoke fetchAllDocuments() method.</li><li>2. Verify the data structure returned.</li><li>3. Check that all document records are returned with correct data.</li></ol>
Expected Results	<ul style="list-style-type: none"><li>- All document records are fetched successfully.</li><li>- Data returned matches the data in the database.</li><li>- Console prints "Total documents fetched: X."</li></ul>
Priority	High
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"><li>- All document records are fetched successfully.</li><li>- Data matches the database.</li><li>- Console outputs "Total documents fetched: X."</li></ul> <p>Fail if:</p> <ul style="list-style-type: none"><li>- Records are not fetched.</li><li>- Data is incorrect or incomplete.</li><li>- Error messages are displayed.</li></ul>

ID	UT13
Title	Fetch Operations - Profiles
Test Description	Verify that all profiles are fetched correctly from the database.
Pre-conditions	<ul style="list-style-type: none"><li>- Database contains at least one profile.</li><li>- Database connection is open.</li></ul>
Test Steps	<ol style="list-style-type: none"><li>1. Invoke fetchAllProfiles() method.</li><li>2. Verify the data structure returned.</li><li>3. Check that all profiles are returned with correct data.</li></ol>
Expected Results	<ul style="list-style-type: none"><li>- All profiles are fetched successfully.</li><li>- Data returned matches the data in the database.</li><li>- Console prints "Successfully retrieved all profiles."</li></ul>
Priority	High
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"><li>- All profiles are fetched successfully.</li><li>- Data matches the database.</li><li>- Console outputs "Successfully retrieved all profiles."</li></ul> <p>Fail if:</p> <ul style="list-style-type: none"><li>- Profiles are not fetched.</li><li>- Data is incorrect or incomplete.</li><li>- Error messages are displayed.</li></ul>



ID	UT14
Title	Delete Operations - Audios
Test Description	Verify that audio records are deleted correctly from the database.
Pre-conditions	<ul style="list-style-type: none"><li>- An existing audio record in the database.</li><li>- Database connection is open.</li></ul>
Test Steps	<ol style="list-style-type: none"><li>1. Fetch an existing audio record's ID.</li><li>2. Invoke deleteAudio() with the ID.</li><li>3. Fetch all audio records using fetchAllAudios().</li><li>4. Verify that the audio record is no longer present.</li></ol>
Expected Results	<ul style="list-style-type: none"><li>- Audio record is deleted successfully.</li><li>- Console prints "Successfully deleted row."</li><li>- Fetching records no longer returns the deleted audio.</li></ul>
Priority	Medium
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"><li>- The audio record is deleted successfully.</li><li>- Console outputs deletion confirmation.</li><li>- Fetching records no longer returns the audio.</li></ul> <p>Fail if:</p> <ul style="list-style-type: none"><li>- Record is not deleted.</li><li>- Deleted record still appears.</li><li>- Error messages are displayed.</li></ul>

ID	UT15
Title	Delete Operations - Documents
Test Description	Verify that document records are deleted correctly from the database.
Pre-conditions	<ul style="list-style-type: none"><li>- An existing document record in the database.</li><li>- Database connection is open.</li></ul>
Test Steps	<ol style="list-style-type: none"><li>1. Fetch an existing document record's ID.</li><li>2. Invoke deleteDocument() with the ID.</li><li>3. Fetch all document records using fetchAllDocuments().</li><li>4. Verify that the document record is no longer present.</li></ol>
Expected Results	<ul style="list-style-type: none"><li>- Document record is deleted successfully.</li><li>- Console prints "Successfully deleted row."</li><li>- Fetching records no longer returns the deleted document.</li></ul>
Priority	Medium
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"><li>- The document record is deleted successfully.</li><li>- Console outputs deletion confirmation.</li><li>- Fetching records no longer returns the document.</li></ul> <p>Fail if:</p> <ul style="list-style-type: none"><li>- Record is not deleted.</li><li>- Deleted record still appears.</li><li>- Error messages are displayed.</li></ul>

ID	UT16
Title	Delete Operations - Profiles
Test Description	Verify that profile records are deleted correctly from the database.
Pre-conditions	<ul style="list-style-type: none"><li>- An existing profile in the database.</li><li>- Database connection is open.</li></ul>
Test Steps	<ol style="list-style-type: none"><li>1. Fetch an existing profile's ProfileName.</li><li>2. Invoke deleteProfile() with the ProfileName.</li><li>3. Fetch all profiles using fetchAllProfiles().</li><li>4. Verify that the profile is no longer present.</li></ol>
Expected Results	<ul style="list-style-type: none"><li>- Profile is deleted successfully.</li><li>- Console prints "Successfully deleted profile."</li><li>- Fetching profiles no longer returns the deleted profile.</li></ul>
Priority	Medium
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"><li>- The profile is deleted successfully.</li><li>- Console outputs deletion confirmation.</li><li>- Fetching records no longer returns the profile.</li></ul> <p>Fail if:</p> <ul style="list-style-type: none"><li>- Record is not deleted.</li><li>- Deleted record still appears.</li><li>- Error messages are displayed.</li></ul>

ID	UT17
Title	Edge Cases - Non-existent Record Deletion
Test Description	Verify that methods handle deletion attempts for non-existent records gracefully.
Pre-conditions	<ul style="list-style-type: none"> <li>- Database connection is open.</li> <li>- No record exists with the specified ID or name.</li> </ul>
Test Steps	<ol style="list-style-type: none"> <li>1. Attempt to delete an audio record with a non-existent ID using deleteAudio().</li> <li>2. Attempt to delete a document record with a non-existent ID using deleteDocument().</li> <li>3. Attempt to delete a profile with a non-existent name using deleteProfile().</li> <li>4. Observe console output for error handling.</li> </ol>
Expected Results	<ul style="list-style-type: none"> <li>- Methods handle non-existent records gracefully.</li> <li>- No unhandled exceptions occur.</li> <li>- Console may print "Could not delete row/profile."</li> <li>- Database remains unaffected.</li> </ul>
Priority	Low
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"> <li>- Methods handle non-existent records gracefully.</li> <li>- No unhandled exceptions occur.</li> <li>- Console prints appropriate error messages.</li> <li>- Database remains unaffected.</li> </ul> <p>Fail if:</p> <ul style="list-style-type: none"> <li>- Unhandled exceptions occur.</li> <li>- Console does not display error messages.</li> <li>- Database is incorrectly modified.</li> </ul>

ID	UT18
Title	Edge Cases - Empty Inserts
Test Description	Verify that database constraints prevent inserting records with empty fields.
Pre-conditions	- Database connection is open.
Test Steps	<ol style="list-style-type: none"><li>1. Attempt to insert an audio record with empty required fields using insertAudio().</li><li>2. Attempt to insert a document record with empty required fields using insertDocument().</li><li>3. Attempt to insert a profile with empty required fields using saveProfile().</li><li>4. Observe the console output for error handling.</li></ol>
Expected Results	<ul style="list-style-type: none"><li>- Database enforces constraints and prevents empty inserts.</li><li>- Appropriate error messages are displayed.</li><li>- No records are inserted with incomplete data.</li></ul>
Priority	Medium
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"><li>- Database constraints prevent empty inserts.</li><li>- Error messages are displayed.</li><li>- No records are inserted.</li></ul> <p>Fail if:</p> <ul style="list-style-type: none"><li>- Records with empty fields are inserted.</li><li>- Constraints are not enforced.</li><li>- Error messages are not displayed.</li></ul>

ID	UT19
Title	Edge Cases - Database Not Initialized
Test Description	Verify that methods handle operations attempted on an uninitialized database gracefully.
Pre-conditions	- Do not initialize the database connection.
Test Steps	1. Attempt to perform database operations (insert, fetch, delete) without initializing the Database class. 2. Observe the console output and application behavior.
Expected Results	- Methods handle the uninitialized database gracefully. - Appropriate error messages are displayed. - No operations are performed on the database.
Priority	High
Pass/Fail	Pass if: - Methods handle uninitialized database gracefully. - Error messages are displayed. - No operations are performed.  Fail if: - Unhandled exceptions occur. - Operations are attempted on an uninitialized database. - Error messages are not displayed.

D	UT20
Title	Linking Records - Audio to Document
Test Description	Verify that an audio record can be linked to a document record correctly.
Pre-conditions	<ul style="list-style-type: none"><li>- Existing audio and document records in the database.</li><li>- Database connection is open.</li></ul>
Test Steps	<ol style="list-style-type: none"><li>1. Fetch an existing audio record's ID.</li><li>2. Fetch an existing document record's ID.</li><li>3. Invoke linkAudioToDocument() with the audio ID and document ID.</li><li>4. Fetch the audio record and verify the DocumentId field is updated.</li></ol>
Expected Results	<ul style="list-style-type: none"><li>- Audio record's DocumentId field is updated with the correct document ID.</li><li>- Console prints "Successfully linked audio to document."</li><li>- Data integrity is maintained.</li></ul>
Priority	Medium
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"><li>- Audio record's DocumentId is updated with the correct ID.</li><li>- Console outputs linking confirmation.</li><li>- Data integrity is maintained.</li></ul> <p>Fail if:</p> <ul style="list-style-type: none"><li>- DocumentId is not updated.</li><li>- Incorrect document ID is linked.</li><li>- Error messages are displayed.</li></ul>

ID	UT21
Title	Linking Records - Invalid Document ID
Test Description	Verify that linking an audio record to an invalid document ID is handled gracefully.
Pre-conditions	<ul style="list-style-type: none"><li>- Existing audio record in the database.</li><li>- Database connection is open.</li></ul>
Test Steps	<ol style="list-style-type: none"><li>1. Fetch an existing audio record's ID.</li><li>2. Use a non-existent document ID.</li><li>3. Invoke linkAudioToDocument() with the audio ID and invalid document ID.</li><li>4. Observe the console output and check for error handling.</li></ol>
Expected Results	<ul style="list-style-type: none"><li>- Method handles invalid document ID gracefully.</li><li>- Console prints "Could not link audio to document."</li><li>- No changes are made to the audio record's DocumentId field.</li></ul>
Priority	Low
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"><li>- Method handles invalid document ID gracefully.</li><li>- Console outputs an appropriate error message.</li><li>- No changes are made to the DocumentId field.</li></ul> <p>Fail if:</p> <ul style="list-style-type: none"><li>- Unhandled exceptions occur.</li><li>- The DocumentId is incorrectly updated.</li><li>- Error messages are not displayed.</li></ul>



ID	UT22
Title	Profile Management - Rename Profile
Test Description	Verify that profiles can be renamed correctly while maintaining data integrity.
Pre-conditions	<ul style="list-style-type: none"> <li>- Existing profile in the database.</li> <li>- Database connection is open.</li> </ul>
Test Steps	<ol style="list-style-type: none"> <li>1. Fetch an existing profile's ProfileName.</li> <li>2. Choose a new unique name for the profile.</li> <li>3. Invoke renameProfile() with the old name and new name.</li> <li>4. Verify that the profile's name is updated.</li> <li>5. Attempt to rename the profile to a duplicate name.</li> <li>6. Observe error handling for duplicate names.</li> </ol>
Expected Results	<ul style="list-style-type: none"> <li>- Profile is renamed successfully to a unique name.</li> <li>- Console prints "Successfully renamed profile."</li> <li>- Renaming to a duplicate name is handled gracefully with an error message.</li> <li>- Database integrity is maintained.</li> </ul>
Priority	Medium
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"> <li>- Profile is renamed successfully to a unique name.</li> <li>- Renaming to a duplicate name is handled gracefully.</li> <li>- Console outputs appropriate messages.</li> <li>- Database integrity is maintained.</li> </ul> <p>Fail if:</p> <ul style="list-style-type: none"> <li>- Profile is not renamed.</li> <li>- Duplicate names are allowed.</li> <li>- Error messages are not displayed.</li> </ul>

ID	UT23
Title	Profile Management - Load Profile by Name
Test Description	Verify that profiles can be loaded by name and match database records accurately.
Pre-conditions	<ul style="list-style-type: none"><li>- Existing profile in the database.</li><li>- Database connection is open.</li></ul>
Test Steps	<ol style="list-style-type: none"><li>1. Invoke loadProfile() with a valid ProfileName.</li><li>2. Verify that the returned profile data matches the database record.</li></ol>
Expected Results	<ul style="list-style-type: none"><li>- Profile is loaded successfully.</li><li>- Returned data matches the database.</li><li>- Console prints "Successfully loaded profile: [ProfileName]."</li></ul>
Priority	High
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"><li>- Profile is loaded successfully.</li><li>- Returned data matches the database.</li><li>- Console outputs the load confirmation.</li></ul> <p>Fail if:</p> <ul style="list-style-type: none"><li>- Profile is not loaded.</li><li>- Data is incorrect.</li><li>- Error messages are not displayed.</li></ul>

ID	UT24
Title	Profile Management - Profile Not Found
Test Description	Verify that the system handles non-existent profiles gracefully.
Pre-conditions	- Database connection is open.
Test Steps	1. Invoke loadProfile() with a ProfileName that does not exist. 2. Observe the console output and method return value.
Expected Results	- Method handles non-existent profiles gracefully. - Console prints "Profile not found." - Method returns nil or an appropriate indication of absence.
Priority	Low
Pass/Fail	Pass if: - Method handles non-existent profiles gracefully. - Console outputs appropriate messages. - Method returns nil or appropriate indication.  Fail if: - Unhandled exceptions occur. - Incorrect data is returned. - Error messages are not displayed.

ID	UT25
Title	Audio Length Calculation
Test Description	Verify that the length of audio files is calculated accurately.
Pre-conditions	<ul style="list-style-type: none"><li>- Valid audio file exists at a specified path.</li><li>- Database connection is open.</li></ul>
Test Steps	<ol style="list-style-type: none"><li>1. Invoke getAudioLength() with the valid audio file path.</li><li>2. Wait for the asynchronous completion handler.</li><li>3. Verify that the returned length is accurate.</li></ol>
Expected Results	<ul style="list-style-type: none"><li>- Audio length is calculated correctly in seconds.</li><li>- Console prints "Audio Length: X seconds."</li><li>- Length is at least 1 second, as per the method logic.</li></ul>
Priority	Medium
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"><li>- Audio length is calculated accurately.</li><li>- Console outputs the audio length.</li><li>- Length meets the expected minimum.</li></ul> <p>Fail if:</p> <ul style="list-style-type: none"><li>- Incorrect length is returned.</li><li>- Console does not display the length.</li><li>- Error messages are displayed.</li></ul>

ID	UT26
Title	Audio Length Calculation - Invalid Path
Test Description	Verify that invalid audio file paths are handled gracefully.
Pre-conditions	<ul style="list-style-type: none"><li>- Invalid or non-existent audio file path.</li><li>- Database connection is open.</li></ul>
Test Steps	<ol style="list-style-type: none"><li>1. Invoke getAudioLength() with an invalid file path.</li><li>2. Wait for the asynchronous completion handler.</li><li>3. Observe error handling and console output.</li></ol>
Expected Results	<ul style="list-style-type: none"><li>- Method handles invalid paths gracefully.</li><li>- Console prints "Failed to load duration with error: [error message]."</li><li>- Completion handler returns nil for length.</li></ul>
Priority	Low
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"><li>- Method handles invalid paths gracefully.</li><li>- Console outputs an appropriate error message.</li><li>- Completion handler returns nil.</li></ul> <p>Fail if:</p> <ul style="list-style-type: none"><li>- Unhandled exceptions occur.</li><li>- Incorrect length is returned.</li><li>- Error messages are not displayed.</li></ul>

ID	UT27
Title	Concurrency Handling
Test Description	Verify that the database maintains data consistency during concurrent operations.
Pre-conditions	<ul style="list-style-type: none"><li>- Database contains multiple records.</li><li>- Database connection is open.</li></ul>
Test Steps	<ol style="list-style-type: none"><li>1. Simulate concurrent access by performing multiple database operations simultaneously (e.g., inserts, updates, deletes).</li><li>2. Use threads or asynchronous tasks to perform operations concurrently.</li><li>3. Observe data consistency and integrity after operations.</li></ol>
Expected Results	<ul style="list-style-type: none"><li>- Database maintains data consistency across concurrent operations.</li><li>- No data corruption occurs.</li><li>- Operations complete without deadlocks or exceptions.</li></ul>
Priority	High
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"><li>- Data consistency is maintained.</li><li>- No data corruption occurs.</li><li>- Operations complete successfully without deadlocks or exceptions.</li></ul> <p>Fail if:</p> <ul style="list-style-type: none"><li>- Data corruption occurs.</li><li>- Deadlocks or exceptions are encountered.</li><li>- Operations fail to complete.</li></ul>

**1.2 TextParser Class Unit Test Items**

ID	UT28
Title	Initialization
Test Description	Verify that a TextParser instance initializes correctly with a valid InputStream.
Pre-conditions	- An InputStream with valid data is available.
Test Steps	1. Create an instance of InputStream with valid data (e.g., a text file or data stream). 2. Initialize a TextParser instance using the InputStream. 3. Observe if the TextParser instance is created successfully without any errors or exceptions.
Expected Results	- TextParser initializes successfully. - No errors or exceptions are thrown during initialization. - Internal properties of TextParser (like inputStream, terminateFlag, totalChunks, totalWordCount) are set correctly.
Priority	High
Pass/Fail	Pass if: - TextParser is initialized without errors. - Internal properties are correctly assigned.  Fail if: - Errors occur during initialization. - Internal properties are not set correctly.

ID	UT29
Title	Start and Stop Parsing - Start Parsing on Background Thread
Test Description	Verify that parsing begins on a background thread and does not block the main thread.
Pre-conditions	<ul style="list-style-type: none"><li>- TextParser instance is initialized with a valid InputStream.</li><li>- A delegate is set for the TextParser instance.</li></ul>
Test Steps	<ol style="list-style-type: none"><li>1. Implement a delegate that can track thread execution.</li><li>2. Assign the delegate to the TextParser instance.</li><li>3. Invoke startParsing() on the TextParser.</li><li>4. Verify that parsing begins on a background thread by checking the thread in delegate callbacks.</li><li>5. Confirm that the main thread remains responsive.</li></ol>
Expected Results	<ul style="list-style-type: none"><li>- Parsing starts on a background thread.</li><li>- Main thread is not blocked.</li><li>- Delegate methods (didProduceChunk, didFinish, etc.) are called appropriately.</li></ul>
Priority	High
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"><li>- Parsing begins on a background thread.</li><li>- Main thread remains responsive.</li><li>- Delegate methods are invoked as expected.</li></ul> <p>Fail if:</p> <ul style="list-style-type: none"><li>- Parsing blocks the main thread.</li><li>- Delegate methods are not called.</li><li>- Errors occur during parsing.</li></ul>



ID	UT30
Title	Start and Stop Parsing - Stop Parsing
Test Description	Verify that parsing stops gracefully when stopParsing() is invoked.
Pre-conditions	- TextParser instance is initialized and parsing has started.
Test Steps	<ol style="list-style-type: none"><li>1. Start parsing by invoking startParsing() on the TextParser.</li><li>2. Allow parsing to proceed for a short duration.</li><li>3. Invoke stopParsing() to set terminateFlag to true.</li><li>4. Observe if parsing stops gracefully.</li><li>5. Verify that textParserDidTerminate delegate method is called with the correct word count.</li></ol>
Expected Results	<ul style="list-style-type: none"><li>- Parsing stops when stopParsing() is called.</li><li>- terminateFlag is set to true.</li><li>- Delegate method textParserDidTerminate is invoked with the total word count parsed until termination.</li><li>- No errors or exceptions occur during termination.</li></ul>
Priority	High
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"><li>- Parsing stops upon calling stopParsing().</li><li>- terminateFlag is correctly set.</li><li>- Delegate method is invoked with accurate word count.</li></ul> <p>Fail if:</p> <ul style="list-style-type: none"><li>- Parsing does not stop.</li><li>- terminateFlag is not set.</li><li>- Delegate method is not called or has incorrect data.</li></ul>

ID	UT31
Title	Chunk Emission - First Chunk
Test Description	Verify that the first chunk emitted follows the specified rules.
Pre-conditions	<ul style="list-style-type: none"><li>- TextParser instance is initialized with a valid InputStream.</li><li>- Delegate is set to receive chunks.</li></ul>
Test Steps	<ol style="list-style-type: none"><li>1. Input text that includes commas, periods, and more than 50 words.</li><li>2. Start parsing using startParsing().</li><li>3. Observe the first chunk emitted by the delegate method didProduceChunk.</li><li>4. Verify that the first chunk ends at the first comma, period, or after 50 words, whichever comes first.</li></ol>
Expected Results	<ul style="list-style-type: none"><li>- First chunk is emitted correctly according to the rules: ends at first comma, period, or 50 words.</li><li>- Chunk contains expected text content.</li><li>- Delegate method is called with the correct chunk.</li></ul>
Priority	Medium
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"><li>- First chunk ends at the correct delimiter or word limit.</li><li>- Chunk content matches expectations.</li></ul> <p>Fail if:</p> <ul style="list-style-type: none"><li>- First chunk does not follow rules.</li><li>- Incorrect chunk content is emitted.</li></ul>

ID	UT32
Title	Chunk Emission - Subsequent Chunks
Test Description	Verify that subsequent chunks are emitted according to specified rules.
Pre-conditions	<ul style="list-style-type: none"> <li>- TextParser instance is initialized with a valid InputStream.</li> <li>- Delegate is set to receive chunks.</li> </ul>
Test Steps	<ol style="list-style-type: none"> <li>1. Input text requiring multiple chunks (e.g., a long text without frequent punctuation).</li> <li>2. Start parsing using startParsing().</li> <li>3. Observe subsequent chunks emitted after the first chunk.</li> <li>4. Verify that chunks are emitted according to the set word limits: <ul style="list-style-type: none"> <li>- Second chunk up to 100 words, ending at comma or period if possible.</li> <li>- Third and subsequent chunks up to 300 words, ending at comma or period if possible.</li> </ul> </li> </ol>
Expected Results	<ul style="list-style-type: none"> <li>- Chunks are emitted correctly according to word limits and punctuation rules.</li> <li>- Each chunk's word count adheres to the limits.</li> <li>- Delegate methods are called with correct chunk data.</li> </ul>
Priority	Medium
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"> <li>- Chunks follow word limits and punctuation rules.</li> <li>- Chunk contents are accurate.</li> </ul> <p>Fail if:</p> <ul style="list-style-type: none"> <li>- Chunks exceed limits or fail to adhere to rules.</li> <li>- Incorrect chunk segmentation occurs.</li> </ul>

ID	UT33
Title	Delegate Call - Parsing Completion
Test Description	Verify that textParserDidFinish delegate method is called upon parsing completion.
Pre-conditions	<ul style="list-style-type: none"><li>- TextParser instance is initialized with a valid InputStream.</li><li>- Delegate is set to receive parsing completion notifications.</li></ul>
Test Steps	<ol style="list-style-type: none"><li>1. Start parsing using startParsing().</li><li>2. Allow parsing to complete naturally (do not terminate early).</li><li>3. Observe if the delegate method textParserDidFinish is called.</li><li>4. Verify that totalChunks matches the actual number of chunks emitted.</li></ol>
Expected Results	<ul style="list-style-type: none"><li>- textParserDidFinish is called upon parsing completion.</li><li>- totalChunks accurately reflects the number of chunks processed.</li><li>- No errors occur during completion callback.</li></ul>
Priority	High
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"><li>- Parsing completes and delegate method is called.</li><li>- totalChunks is accurate.</li></ul> <p>Fail if:</p> <ul style="list-style-type: none"><li>- Delegate method is not called.</li><li>- totalChunks is incorrect.</li></ul>

ID	UT34
Title	Delegate Call - Parsing Termination
Test Description	Verify that textParserDidTerminate delegate method is called upon parsing termination.
Pre-conditions	<ul style="list-style-type: none"><li>- TextParser instance is initialized with a valid InputStream.</li><li>- Delegate is set to receive termination notifications.</li></ul>
Test Steps	<ol style="list-style-type: none"><li>1. Start parsing using startParsing().</li><li>2. After parsing some data, invoke stopParsing() to terminate.</li><li>3. Observe if the delegate method textParserDidTerminate is called.</li><li>4. Verify that wordCount matches the actual number of words parsed before termination.</li></ol>
Expected Results	<ul style="list-style-type: none"><li>- textParserDidTerminate is called upon termination.</li><li>- wordCount accurately reflects the number of words parsed before termination.</li><li>- Parsing stops gracefully without errors.</li></ul>
Priority	High
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"><li>- Parsing terminates and delegate method is called.</li><li>- wordCount is accurate.</li></ul> <p>Fail if:</p> <ul style="list-style-type: none"><li>- Delegate method is not called.</li><li>- wordCount is incorrect.</li></ul>

**1.3 PasswordManager Class Unit Test Items**

ID	UT35
Title	Initialization
Test Description	Verify that the PasswordManager initializes correctly when no password or recovery code is stored.
Pre-conditions	- No password or recovery code is stored in UserDefaults.
Test Steps	1. Ensure UserDefaults does not contain values for "appPassword" and "recoveryCode". 2. Initialize an instance of PasswordManager. 3. Observe the values of hasPassword and recoveryCode.
Expected Results	- hasPassword is false. - recoveryCode is nil. - No errors or exceptions occur during initialization.
Priority	High
Pass/Fail	Pass if: - hasPassword is false. - recoveryCode is nil. - No errors or exceptions occur.  Fail if: - hasPassword is true. - recoveryCode is not nil. - Errors occur during initialization.

ID	UT36
Title	Set Password
Test Description	Verify that a password is set correctly and a recovery code is generated.
Pre-conditions	<ul style="list-style-type: none"> <li>- No password is currently set in UserDefaults.</li> <li>- PasswordManager is initialized.</li> </ul>
Test Steps	<ol style="list-style-type: none"> <li>1. Call setPassword("TestPassword123") on the PasswordManager instance.</li> <li>2. Capture the returned recovery code.</li> <li>3. Verify the values of hasPassword and recoveryCode.</li> <li>4. Check that the password is stored in UserDefaults under "appPassword".</li> <li>5. Verify the recovery code is stored under "recoveryCode".</li> </ol>
Expected Results	<ul style="list-style-type: none"> <li>- Password is stored in UserDefaults.</li> <li>- hasPassword is true.</li> <li>- A recovery code is generated, returned, and stored.</li> <li>- recoveryCode matches the stored value.</li> </ul>
Priority	High
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"> <li>- Password is stored.</li> <li>- hasPassword is true.</li> <li>- Recovery code is generated, returned, and stored.</li> </ul> <p>Fail if:</p> <ul style="list-style-type: none"> <li>- Password is not stored.</li> <li>- hasPassword is not updated.</li> <li>- Recovery code is not generated or stored.</li> </ul>

ID	UT37
Title	Remove Password
Test Description	Verify that a password and recovery code can be removed successfully.
Pre-conditions	<ul style="list-style-type: none"><li>- A password and recovery code are set in UserDefaults.</li><li>- PasswordManager is initialized with hasPassword being true.</li></ul>
Test Steps	<ol style="list-style-type: none"><li>1. Call removePassword() on the PasswordManager instance.</li><li>2. Observe the values of hasPassword and recoveryCode.</li><li>3. Verify that "appPassword" and "recoveryCode" keys are removed from UserDefaults.</li></ol>
Expected Results	<ul style="list-style-type: none"><li>- Password and recovery code are removed.</li><li>- hasPassword is false.</li><li>- recoveryCode is nil.</li></ul>
Priority	Medium
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"><li>- Password and recovery code are removed.</li><li>- hasPassword is false.</li><li>- recoveryCode is nil.</li></ul> <p>Fail if:</p> <ul style="list-style-type: none"><li>- Values remain in UserDefaults.</li><li>- hasPassword is not updated.</li><li>- Errors occur.</li></ul>



ID	UT38
Title	Validate Password
Test Description	Verify that password validation works correctly.
Pre-conditions	<ul style="list-style-type: none"><li>- A password ("TestPassword123") is stored in UserDefaults.</li><li>- PasswordManager is initialized.</li></ul>
Test Steps	<ol style="list-style-type: none"><li>1. Call validatePassword("TestPassword123").</li><li>2. Observe the return value.</li><li>3. Call validatePassword("WrongPassword").</li><li>4. Observe the return value.</li></ol>
Expected Results	<ul style="list-style-type: none"><li>- Validation with the correct password returns true.</li><li>- Validation with an incorrect password returns false.</li></ul>
Priority	High
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"><li>- Correct password validation returns true.</li><li>- Incorrect password validation returns false.</li></ul> <p>Fail if:</p> <ul style="list-style-type: none"><li>- Correct validation fails.</li><li>- Incorrect validation succeeds.</li></ul>

ID	UT39
Title	Recover Account
Test Description	Verify that the recovery process works as expected.
Pre-conditions	<ul style="list-style-type: none"> <li>- A password and recovery code are stored in UserDefaults.</li> <li>- PasswordManager is initialized.</li> </ul>
Test Steps	<ol style="list-style-type: none"> <li>1. Retrieve the stored recovery code from UserDefaults.</li> <li>2. Call recoverAccount(with: recoveryCode, newPassword: "NewPassword123").</li> <li>3. Verify the password is updated.</li> <li>4. Verify the recovery code is removed.</li> <li>5. Call recoverAccount(with: "InvalidCode", newPassword: "AnotherPassword").</li> <li>6. Verify the return value and password remain unchanged.</li> </ol>
Expected Results	<ul style="list-style-type: none"> <li>- Recovery with the correct code updates the password and removes the recovery code.</li> <li>- Recovery with an invalid code fails and leaves the password unchanged.</li> </ul>
Priority	High
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"> <li>- Recovery succeeds with the correct code.</li> <li>- Password is updated.</li> <li>- Recovery code is cleared.</li> </ul> <p>Fail if:</p> <ul style="list-style-type: none"> <li>- Recovery fails with the correct code.</li> <li>- Password is not updated after successful recovery.</li> <li>- Recovery succeeds with an invalid code.</li> </ul>

ID	UT40
Title	Generate Recovery Code
Test Description	Verify that recovery codes are generated correctly.
Pre-conditions	- N/A (method is private, tested via methods that use it).
Test Steps	1. Call setPassword("TestPassword123") to generate a recovery code. 2. Retrieve the recovery code. 3. Verify it is a six-digit numeric string within "000000" to "999999".
Expected Results	- Recovery code is a six-digit numeric string within the expected range. - No errors occur during generation.
Priority	Medium
Pass/Fail	Pass if: - Recovery code is six digits. - Code is numeric and within range.  Fail if: - Code is not six digits. - Contains non-numeric characters. - Errors occur.

ID	UT41
Title	Finalize New Password
Test Description	Verify that the temporary password is replaced correctly with a new password.
Pre-conditions	<ul style="list-style-type: none"> <li>- temporaryPassword is set (simulate via generateTemporaryRecoveryCode).</li> <li>- PasswordManager is initialized.</li> </ul>
Test Steps	<ol style="list-style-type: none"> <li>1. Call generateTemporaryRecoveryCode(for: "TempPassword123").</li> <li>2. Retrieve the recovery code.</li> <li>3. Call finalizeNewPassword("TempPassword123").</li> <li>4. Verify that hasPassword is true.</li> <li>5. Check that the new password is stored.</li> <li>6. Call finalizeNewPassword("WrongPassword").</li> <li>7. Verify that the password is not updated.</li> </ol>
Expected Results	<ul style="list-style-type: none"> <li>- Correct temporary password sets the new password.</li> <li>- hasPassword is true.</li> <li>- Password is stored in UserDefaults.</li> <li>- Incorrect password does not update the stored password.</li> </ul>
Priority	Medium
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"> <li>- Correct temporary password sets the new password.</li> <li>- hasPassword is true.</li> <li>- Incorrect password does not update the stored password.</li> </ul> <p>Fail if:</p> <ul style="list-style-type: none"> <li>- Correct temporary password fails.</li> <li>- Incorrect password updates the stored password.</li> </ul>

### 1.4 CombineAudioFiles Function Unit Test Items

ID	UT42
Item to Test	Stitch Audios
Test Description	Verify that multiple audio files can be combined into a single file correctly.
Pre-Conditions	<ul style="list-style-type: none"> <li>- Multiple valid audio files are available at specified URLs.</li> <li>- The audio files are in a compatible format (e.g., WAV files).</li> <li>- The output directory ("audiofiles" in the application support directory) is accessible and writable.</li> <li>- The FileDirectory class is functional.</li> </ul>
Test Steps	<ol style="list-style-type: none"> <li>1. Prepare Input Audio Files: Ensure at least two valid audio files (e.g., audio1.wav, audio2.wav) exist.</li> <li>2. Obtain File URLs for the audio files.</li> <li>3. Invoke combineAudioFiles with the URLs and an output file name ("CombinedAudioTest").</li> <li>4. Handle Completion: Check if the result is .success or .failure.</li> <li>5. Verify Output File Creation: Ensure the combined audio file is created in the output directory.</li> <li>6. Playback Combined Audio: Use AVAudioPlayer to play the combined audio.</li> <li>7. Verify Audio Content: Listen to ensure correct order and smooth playback.</li> <li>8. Check Duration: Verify the combined audio duration matches the sum of the individual file durations.</li> <li>9. Cleanup: Remove the combined file if necessary.</li> </ol>
Expected Results	<ul style="list-style-type: none"> <li>- The combineAudioFiles function completes successfully without errors.</li> <li>- The combined file is created in the audiofiles directory.</li> <li>- The audio is combined correctly in the correct order.</li> <li>- Playback is smooth without glitches.</li> <li>- The duration matches the sum of input audio files.</li> <li>- No exceptions or unhandled errors occur.</li> </ul>
Priority	High
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"> <li>- The function executes without errors.</li> <li>- The audio file is created and correct.</li> <li>- The duration is accurate.</li> </ul> <p>Fail if:</p> <ul style="list-style-type: none"> <li>- Errors occur.</li> <li>- The combined audio file is corrupted.</li> <li>- Duration mismatch.</li> </ul>

EchoText

**1.5 TtsLogic Class Unit Test Items**

ID	UT43
Item to Test	Initialization
Test Description	Verify that TtsLogic initializes successfully with the provided dependencies.
Pre-Conditions	<ul style="list-style-type: none"><li>- A Database instance is available.</li><li>- A FileDirectory instance is available.</li><li>- A VoiceProfileManager instance is available.</li></ul>
Test Steps	<ol style="list-style-type: none"><li>1. Initialize a TtsLogic instance with the Database, FileDirectory, and VoiceProfileManager.</li><li>2. Observe if the instance is created successfully.</li><li>3. Verify that internal properties (e.g., isGenerating, audioQueuePlayer) are set correctly.</li></ol>
Expected Results	<ul style="list-style-type: none"><li>- TtsLogic initializes without errors.</li><li>- Internal properties are set to default values.</li><li>- No errors or exceptions occur.</li></ul>
Priority	High
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"><li>- TtsLogic is initialized without errors.</li><li>- Internal properties are correctly assigned.</li></ul> <p>Fail if:</p> <ul style="list-style-type: none"><li>- Errors occur during initialization.</li><li>- Internal properties are incorrect.</li></ul>

ID	UT44
Item to Test	Generate Speech Streaming
Test Description	Verify that text input is correctly processed and speech is generated in chunks.
Pre-Conditions	<ul style="list-style-type: none"><li>- TtsLogic instance is initialized.</li><li>- Text input is provided.</li><li>- VoiceProfileManager is configured.</li></ul>
Test Steps	<ol style="list-style-type: none"><li>1. Set TtsLogic.text to a valid string (e.g., "This is a test").</li><li>2. Call generateSpeechStreaming() on the instance.</li><li>3. Observe if text is processed into chunks.</li><li>4. Verify TTS generation for each chunk.</li><li>5. Monitor delegate callbacks and state changes.</li></ol>
Expected Results	<ul style="list-style-type: none"><li>- Text is parsed into chunks.</li><li>- TTS generation occurs for each chunk.</li><li>- Audio chunks are processed without errors.</li></ul>
Priority	High
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"><li>- Text is correctly chunked.</li><li>- TTS generation occurs for each chunk.</li><li>- Audio chunks are processed.</li></ul> <p>Fail if:</p> <ul style="list-style-type: none"><li>- Text is not chunked.</li><li>- TTS generation fails.</li><li>- Errors occur during processing.</li></ul>



ID	UT45
Item to Test	Stop Speech
Test Description	Verify that speech generation stops correctly when stopSpeech() is invoked.
Pre-Conditions	<ul style="list-style-type: none"><li>- TtsLogic instance is initialized.</li><li>- Speech generation is in progress.</li></ul>
Test Steps	<ol style="list-style-type: none"><li>1. Call stopSpeech() while speech generation is active.</li><li>2. Verify that text parsing halts.</li><li>3. Check that audio queues are cleared.</li><li>4. Ensure generation states reset (e.g., isGenerating is false).</li></ol>
Expected Results	<ul style="list-style-type: none"><li>- Speech generation stops immediately.</li><li>- Text parsing halts.</li><li>- Audio queues are cleared.</li><li>- Internal states are reset appropriately.</li></ul>
Priority	High
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"><li>- Speech stops when stopSpeech() is called.</li><li>- Parsing halts.</li><li>- Queues and states reset.</li></ul> <p>Fail if:</p> <ul style="list-style-type: none"><li>- Speech does not stop.</li><li>- Queues are not cleared.</li><li>- States are not reset.</li></ul>

ID	UT46
Item to Test	Audio Chunk Handling
Test Description	Verify that audio chunks are handled correctly during speech generation.
Pre-Conditions	<ul style="list-style-type: none"><li>- TtsLogic instance is initialized.</li><li>- Speech generation is in progress.</li></ul>
Test Steps	<ol style="list-style-type: none"><li>1. Monitor audio chunk handling during generation.</li><li>2. Verify audio chunks are enqueued into AVQueuePlayer.</li><li>3. Check that playback occurs smoothly.</li></ol>
Expected Results	<ul style="list-style-type: none"><li>- Audio chunks are enqueued properly.</li><li>- Playback is smooth and continuous.</li><li>- No errors occur during enqueueing and playback.</li></ul>
Priority	Medium
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"><li>- Audio chunks are enqueued correctly.</li><li>- Playback is smooth.</li></ul> <p>Fail if:</p> <ul style="list-style-type: none"><li>- Audio chunks are not enqueued.</li><li>- Playback errors occur.</li></ul>

ID	UT47
Item to Test	Playback Completion
Test Description	Verify that playback completion is handled correctly.
Pre-Conditions	<ul style="list-style-type: none"><li>- TtsLogic instance is initialized.</li><li>- Speech generation and playback are in progress.</li></ul>
Test Steps	<ol style="list-style-type: none"><li>1. Allow playback to complete naturally.</li><li>2. Observe the playerIdFinishPlaying method handling.</li><li>3. Verify that the audio queue is cleared after playback.</li><li>4. Check that internal states are reset.</li></ol>
Expected Results	<ul style="list-style-type: none"><li>- playerIdFinishPlaying handles playback completion.</li><li>- Audio queue is cleared.</li><li>- Internal states (e.g., isGenerating) are reset.</li></ul>
Priority	Medium
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"><li>- Playback completes without errors.</li><li>- Method is called.</li><li>- Queue cleared, states reset.</li></ul> <p>Fail if:</p> <ul style="list-style-type: none"><li>- Playback completion is not handled.</li><li>- Queue is not cleared.</li><li>- States are not reset.</li></ul>

ID	UT48
Item to Test	Text Parsing Delegate
Test Description	Verify that the text parsing delegate processes chunks correctly.
Pre-Conditions	<ul style="list-style-type: none"><li>- TtsLogic instance is initialized.</li><li>- Text parsing is in progress.</li></ul>
Test Steps	<ol style="list-style-type: none"><li>1. Monitor textParser(_:didProduceChunk:) calls.</li><li>2. Verify chunks are processed correctly for TTS.</li><li>3. Ensure chunks are passed to the TTS system without errors.</li></ol>
Expected Results	<ul style="list-style-type: none"><li>- Delegate method is called for each chunk.</li><li>- Chunks are processed correctly for TTS generation.</li><li>- No errors occur during processing.</li></ul>
Priority	Medium
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"><li>- Delegate is called per chunk.</li><li>- Chunks are processed correctly.</li></ul> <p>Fail if:</p> <ul style="list-style-type: none"><li>- Delegate is not called.</li><li>- Chunks are not processed correctly.</li></ul>

ID	UT49
Item to Test	Text Parsing Completion
Test Description	Verify that the text parsing completion triggers subsequent operations correctly.
Pre-Conditions	<ul style="list-style-type: none"> <li>- TtsLogic instance is initialized.</li> <li>- Text parsing is in progress.</li> </ul>
Test Steps	<ol style="list-style-type: none"> <li>1. Allow parsing to complete.</li> <li>2. Monitor textParserDidFinish(_:totalChunks:).</li> <li>3. Verify audio stitching is triggered upon completion.</li> <li>4. Check that database updates occur as expected.</li> </ol>
Expected Results	<ul style="list-style-type: none"> <li>- Delegate method is called upon completion.</li> <li>- Audio stitching is initiated.</li> <li>- Database is updated with a new audio record.</li> <li>- No errors occur during the process.</li> </ul>
Priority	High
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"> <li>- Delegate is called upon completion.</li> <li>- Stitching and database updates occur.</li> </ul> <p>Fail if:</p> <ul style="list-style-type: none"> <li>- Delegate is not called.</li> <li>- Stitching fails.</li> <li>- Database is not updated.</li> </ul>

ID	UT50
Item to Test	Change Model
Test Description	Verify that the active TTS model can be changed successfully.
Pre-Conditions	<ul style="list-style-type: none"><li>- TtsLogic instance is initialized.</li><li>- No speech generation is in progress (isGenerating is false).</li></ul>
Test Steps	<ol style="list-style-type: none"><li>1. Call changeModel(to:) with a new model name.</li><li>2. Verify that VoiceProfileManager updates the active model.</li><li>3. Ensure the TTS instance is reset or reinitialized.</li><li>4. Check that subsequent TTS operations use the new model.</li></ol>
Expected Results	<ul style="list-style-type: none"><li>- Active model is updated to the new model.</li><li>- TTS instance is reset or reinitialized.</li><li>- No errors occur.</li><li>- Subsequent TTS uses the new model.</li></ul>
Priority	Medium
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"><li>- Model is changed successfully.</li><li>- TTS is updated.</li><li>- No errors occur.</li></ul> <p>Fail if:</p> <ul style="list-style-type: none"><li>- Model is not updated.</li><li>- TTS uses old model.</li><li>- Errors occur during model change.</li></ul>

ID	UT51
Item to Test	Reset
Test Description	Verify that the reset() function clears internal states and reinitializes TTS.
Pre-Conditions	<ul style="list-style-type: none"><li>- TtsLogic instance is initialized.</li><li>- Speech generation may be in progress.</li></ul>
Test Steps	<ol style="list-style-type: none"><li>1. Call reset() on the instance.</li><li>2. Verify that all internal states are cleared.</li><li>3. Ensure that TTS is reinitialized.</li><li>4. Check that progress indicators (e.g., progress, isGenerating) are reset.</li></ol>
Expected Results	<ul style="list-style-type: none"><li>- Internal states are reset.</li><li>- TTS instances are reinitialized.</li><li>- Progress indicators are reset.</li><li>- No errors occur during reset.</li></ul>
Priority	Medium
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"><li>- Reset clears states and reinitializes TTS.</li><li>- Progress indicators are reset.</li></ul> <p>Fail if:</p> <ul style="list-style-type: none"><li>- States are not reset.</li><li>- TTS is not reinitialized.</li><li>- Errors occur during reset.</li></ul>

**1.6 FileDirectory Class Unit Test Items**

ID	UT52
Item to Test	Initialization
Test Description	Verify that the FileDirectory initializes correctly and returns valid URLs for directories.
Pre-Conditions	- N/A.
Test Steps	<ol style="list-style-type: none"><li>1. Create an instance of FileDirectory.</li><li>2. Access the databaseDirectory, documentsDirectory, and audiofilesDirectory properties.</li><li>3. Verify that each property returns a valid URL.</li><li>4. Check that the URLs point to the correct paths in the Application Support directory.</li></ol>
Expected Results	<ul style="list-style-type: none"><li>- FileDirectory initializes successfully.</li><li>- Each directory property returns a valid URL.</li><li>- URLs point to the expected directories.</li><li>- No errors or exceptions occur.</li></ul>
Priority	High
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"><li>- FileDirectory is created without errors.</li><li>- All directory properties return valid URLs.</li></ul> <p>Fail if:</p> <ul style="list-style-type: none"><li>- Initialization fails.</li><li>- Directory properties return invalid or nil URLs.</li></ul>



ID	UT53
Item to Test	Directory Creation
Test Description	Verify that the createApplicationSupportDirectories() method creates the required directories.
Pre-Conditions	<ul style="list-style-type: none"><li>- FileDirectory instance is initialized.</li><li>- The directories do not exist.</li></ul>
Test Steps	<ol style="list-style-type: none"><li>1. Ensure the "database", "documents", and "audiofiles" directories do not exist.</li><li>2. Call createApplicationSupportDirectories().</li><li>3. Check the file system to verify the directories were created.</li><li>4. Verify no errors occurred during creation.</li></ol>
Expected Results	<ul style="list-style-type: none"><li>- The "database", "documents", and "audiofiles" directories are created in the Application Support directory.</li><li>- No errors or exceptions occur.</li></ul>
Priority	High
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"><li>- All three directories are created in the correct location.</li><li>- No errors occur.</li></ul> <p>Fail if:</p> <ul style="list-style-type: none"><li>- Directories are not created.</li><li>- Errors occur during creation.</li></ul>

ID	UT54
Item to Test	Directory Existence
Test Description	Verify that the createApplicationSupportDirectories() method does not modify existing directories.
Pre-Conditions	<ul style="list-style-type: none"><li>- FileDirectory instance is initialized.</li><li>- The directories already exist.</li></ul>
Test Steps	<ol style="list-style-type: none"><li>1. Ensure the "database", "documents", and "audiofiles" directories already exist.</li><li>2. Call createApplicationSupportDirectories().</li><li>3. Observe if any errors occur.</li><li>4. Verify that the existing directories are not modified.</li></ol>
Expected Results	<ul style="list-style-type: none"><li>- The method completes without errors.</li><li>- Existing directories remain intact.</li><li>- No data loss or corruption occurs.</li></ul>
Priority	Medium
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"><li>- Method completes without errors.</li><li>- Existing directories are not modified.</li></ul> <p>Fail if:</p> <ul style="list-style-type: none"><li>- Errors occur.</li><li>- Existing directories are altered.</li></ul>

ID	UT55
Item to Test	Directory Permissions
Test Description	Verify that the application can write to and read from the required directories.
Pre-Conditions	<ul style="list-style-type: none"><li>- FileDirectory instance is initialized.</li><li>- The directories exist.</li></ul>
Test Steps	<ol style="list-style-type: none"><li>1. Ensure the "database", "documents", and "audiofiles" directories exist.</li><li>2. Attempt to write a test file to each directory.</li><li>3. Read back the test file to confirm accessibility.</li><li>4. Delete the test file after the test.</li></ol>
Expected Results	<ul style="list-style-type: none"><li>- The application can write to each directory.</li><li>- Files can be read back from each directory.</li><li>- No permissions errors occur during read/write operations.</li></ul>
Priority	High
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"><li>- Write and read operations succeed.</li><li>- No permissions errors occur.</li></ul> <p>Fail if:</p> <ul style="list-style-type: none"><li>- Unable to write to directories.</li><li>- Permissions errors occur.</li></ul>

ID	UT56
Item to Test	Error Handling
Test Description	Verify that directory creation errors are handled gracefully.
Pre-Conditions	<ul style="list-style-type: none"><li>- FileDirectory instance is initialized.</li><li>- Simulate conditions causing directory creation failure.</li></ul>
Test Steps	<ol style="list-style-type: none"><li>1. Modify the Application Support directory to simulate insufficient permissions (e.g., set to read-only).</li><li>2. Call createApplicationSupportDirectories().</li><li>3. Observe if errors are caught and logged.</li><li>4. Verify the application does not crash and handles the error gracefully.</li></ol>
Expected Results	<ul style="list-style-type: none"><li>- Errors are caught and logged with a meaningful message.</li><li>- The application does not crash.</li><li>- No instability occurs due to unhandled errors.</li></ul>
Priority	Medium
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"><li>- Errors are handled gracefully without crashing.</li><li>- Error messages are logged appropriately.</li></ul> <p>Fail if:</p> <ul style="list-style-type: none"><li>- Application crashes.</li><li>- Errors are not logged or handled.</li></ul>

**1.7 VoiceProfileManager Class Unit Test Items**

ID	UT57
Item to Test	Initialization - Default Profile Creation
Test Description	Verify that a default profile is created if no profiles exist in the database or UserDefaults.
Pre-Conditions	<ul style="list-style-type: none"><li>- No saved profiles exist in the database.</li><li>- UserDefaults does not contain "currentVoiceProfileName".</li></ul>
Test Steps	<ol style="list-style-type: none"><li>1. Ensure the database has no profiles (delete any existing profiles if necessary).</li><li>2. Remove any value associated with "currentVoiceProfileName" in UserDefaults.</li><li>3. Initialize a VoiceProfileManager instance with the database.</li><li>4. Observe the properties of the VoiceProfileManager instance (profileName, pitch, speed, model, id).</li></ol>
Expected Results	<ul style="list-style-type: none"><li>- VoiceProfileManager initializes successfully without errors.</li><li>- A default profile is created with:<ul style="list-style-type: none"><li>• profileName: "Default"</li><li>• pitch: 1.0</li><li>• speed: 1.0</li><li>• model: "amy".</li></ul></li><li>- The default profile is saved in the database.</li><li>- UserDefaults is updated with "currentVoiceProfileName" set to "Default".</li></ul>
Priority	High
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"><li>- Default profile is created and saved with correct values.</li><li>- UserDefaults is updated.</li></ul> <p>Fail if:</p> <ul style="list-style-type: none"><li>- Initialization fails.</li><li>- Default profile is not created or saved.</li><li>- UserDefaults is not updated.</li></ul>

ID	UT58
Item to Test	Initialization - Load Saved Profile
Test Description	Verify that the VoiceProfileManager loads an existing profile from the database and UserDefaults.
Pre-Conditions	<ul style="list-style-type: none"><li>- A profile named "TestProfile" exists in the database with values (pitch: 1.2, speed: 0.8, model: "bob").</li><li>- UserDefaults contains "currentVoiceProfileName" set to "TestProfile".</li></ul>
Test Steps	<ol style="list-style-type: none"><li>1. Ensure the database contains "TestProfile" with the specified values.</li><li>2. Set UserDefaults key "currentVoiceProfileName" to "TestProfile".</li><li>3. Initialize a VoiceProfileManager instance with the database.</li><li>4. Observe the properties of the VoiceProfileManager instance.</li></ol>
Expected Results	<ul style="list-style-type: none"><li>- VoiceProfileManager initializes successfully without errors.</li><li>- The profile "TestProfile" is loaded from the database.</li><li>- Properties match "TestProfile"'s values (pitch: 1.2, speed: 0.8, model: "bob").</li></ul>
Priority	High
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"><li>- "TestProfile" is loaded correctly.</li><li>- Properties match the saved profile.</li><li>- No default profile is created.</li></ul> <p>Fail if:</p> <ul style="list-style-type: none"><li>- Incorrect profile is loaded.</li><li>- Properties do not match the saved values.</li><li>- Default profile is created instead.</li></ul>

ID	UT59
Item to Test	Profile Switching - switchToProfile
Test Description	Verify that switching between profiles saves changes and updates active properties correctly.
Pre-Conditions	<ul style="list-style-type: none"> <li>- Two profiles ("ProfileA" and "ProfileB") exist in the database.</li> <li>- "ProfileA" is the current active profile.</li> <li>- VoiceProfileManager is initialized and active.</li> </ul>
Test Steps	<ol style="list-style-type: none"> <li>1. Modify a property in "ProfileA" (e.g., change pitch from 1.0 to 1.5).</li> <li>2. Call switchToProfile(named: "ProfileB") on the VoiceProfileManager.</li> <li>3. Verify that changes to "ProfileA" are saved in the database.</li> <li>4. Confirm the active profile switches to "ProfileB".</li> <li>5. Check that properties match "ProfileB"'s values.</li> <li>6. Ensure UserDefaults is updated with "currentVoiceProfileName" set to "ProfileB".</li> </ol>
Expected Results	<ul style="list-style-type: none"> <li>- Changes to "ProfileA" are saved before switching.</li> <li>- Active profile switches to "ProfileB".</li> <li>- Properties update to "ProfileB"'s values.</li> <li>- UserDefaults reflects the new active profile.</li> <li>- No errors occur.</li> </ul>
Priority	High
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"> <li>- Changes to "ProfileA" are saved.</li> <li>- Active profile switches to "ProfileB".</li> <li>- Properties match "ProfileB".</li> <li>- UserDefaults is updated.</li> </ul> <p>Fail if:</p> <ul style="list-style-type: none"> <li>- Changes to "ProfileA" are not saved.</li> <li>- Active profile does not switch.</li> <li>- Errors occur during switching.</li> </ul>

ID	UT60
Item to Test	Add Profile - addProfile
Test Description	Verify that a new profile can be added and set as active.
Pre-Conditions	<ul style="list-style-type: none"> <li>- VoiceProfileManager is initialized.</li> <li>- No profile named "NewProfile" exists in the database.</li> </ul>
Test Steps	<ol style="list-style-type: none"> <li>1. Call addProfile(named: "NewProfile") on the VoiceProfileManager.</li> <li>2. Verify that "NewProfile" is created with default values (pitch: 1.0, speed: 1.0, model: "amy").</li> <li>3. Confirm that the active profile switches to "NewProfile".</li> <li>4. Check that "NewProfile" is saved in the database.</li> <li>5. Ensure UserDefaults is updated with "currentVoiceProfileName" set to "NewProfile".</li> </ol>
Expected Results	<ul style="list-style-type: none"> <li>- "NewProfile" is created and saved in the database.</li> <li>- Active profile switches to "NewProfile".</li> <li>- Properties reflect default values.</li> <li>- UserDefaults is updated.</li> <li>- No errors occur.</li> </ul>
Priority	High
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"> <li>- New profile is created and active.</li> <li>- Profile is saved in the database.</li> <li>- UserDefaults reflects the new profile.</li> </ul> <p>Fail if:</p> <ul style="list-style-type: none"> <li>- Profile is not created or saved.</li> <li>- Active profile does not switch.</li> <li>- Errors occur during addition.</li> </ul>



ID	UT61
Item to Test	Rename Profile - renameProfile
Test Description	Verify that a profile's name can be successfully updated and reflected in the system.
Pre-Conditions	<ul style="list-style-type: none"> <li>- A profile named "OldName" exists and is active.</li> <li>- VoiceProfileManager is initialized and active.</li> </ul>
Test Steps	<ol style="list-style-type: none"> <li>1. Call renameProfile(from: "OldName", to: "NewName") on the VoiceProfileManager.</li> <li>2. Verify that the profile's name is updated in the database to "NewName".</li> <li>3. Confirm that profileName in VoiceProfileManager is updated to "NewName".</li> <li>4. Ensure UserDefaults is updated if the active profile was renamed.</li> </ol>
Expected Results	<ul style="list-style-type: none"> <li>- Profile is renamed in the database from "OldName" to "NewName".</li> <li>- Active profile's profileName is updated.</li> <li>- UserDefaults reflects the new profile name.</li> <li>- No errors occur during the process.</li> </ul>
Priority	Medium
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"> <li>- Profile is renamed successfully.</li> <li>- Active profile updates its name.</li> <li>- UserDefaults is updated.</li> </ul> <p>Fail if:</p> <ul style="list-style-type: none"> <li>- Profile is not renamed.</li> <li>- Active profile name is incorrect.</li> <li>- Errors occur during renaming.</li> </ul>

ID	UT62
Item to Test	Delete Profile - deleteProfile
Test Description	Verify that a profile can be deleted and the system updates appropriately.
Pre-Conditions	<ul style="list-style-type: none"> <li>- A profile named "ProfileToDelete" exists and is active.</li> <li>- VoiceProfileManager is initialized.</li> </ul>
Test Steps	<ol style="list-style-type: none"> <li>1. Call deleteProfile(named: "ProfileToDelete") on the VoiceProfileManager.</li> <li>2. Verify that "ProfileToDelete" is removed from the database.</li> <li>3. Confirm that if "ProfileToDelete" was active, the active profile switches to "Default".</li> <li>4. Ensure UserDefaults is updated with "currentVoiceProfileName" set to "Default" if necessary.</li> </ol>
Expected Results	<ul style="list-style-type: none"> <li>- "ProfileToDelete" is deleted from the database.</li> <li>- If active, the profile switches to "Default".</li> <li>- UserDefaults reflects the active profile change.</li> <li>- No errors occur during deletion.</li> </ul>
Priority	Medium
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"> <li>- Profile is deleted from the database.</li> <li>- Active profile updates if necessary.</li> <li>- UserDefaults is updated.</li> </ul> <p>Fail if:</p> <ul style="list-style-type: none"> <li>- Profile is not deleted.</li> <li>- Active profile does not switch when it should.</li> <li>- Errors occur during deletion.</li> </ul>

ID	UT63
Item to Test	Profile Updates via Observers
Test Description	Verify that changes to profile properties are automatically saved to the database via observers.
Pre-Conditions	- VoiceProfileManager is initialized and an active profile is loaded.
Test Steps	<ol style="list-style-type: none"><li>1. Modify pitch, speed, or model properties of the active profile in VoiceProfileManager.</li><li>2. Observe that changes are automatically saved to the database through observers.</li><li>3. Retrieve the profile from the database directly.</li><li>4. Verify that the retrieved profile's values match the updated properties.</li></ol>
Expected Results	<ul style="list-style-type: none"><li>- Changes to properties are saved to the database automatically.</li><li>- Database reflects the updated values without manual saving.</li><li>- No errors occur during the update process.</li></ul>
Priority	High
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"><li>- Property changes are saved automatically.</li><li>- Database values match the updated properties.</li></ul> <p>Fail if:</p> <ul style="list-style-type: none"><li>- Changes are not saved to the database.</li><li>- Database values do not reflect updates.</li><li>- Errors occur during property changes.</li></ul>

ID	UT64
Item to Test	Prevent Default Profile Changes
Test Description	Verify that the "Default" profile cannot be deleted or renamed.
Pre-Conditions	<ul style="list-style-type: none"> <li>- VoiceProfileManager is initialized.</li> <li>- The "Default" profile exists in the database.</li> </ul>
Test Steps	<ol style="list-style-type: none"> <li>1. Attempt to delete the "Default" profile by calling deleteProfile(named: "Default").</li> <li>2. Attempt to rename the "Default" profile by calling renameProfile(from: "Default", to: "NewDefault").</li> <li>3. Observe the return values of both methods.</li> <li>4. Verify that the "Default" profile remains unchanged in the database.</li> </ol>
Expected Results	<ul style="list-style-type: none"> <li>- Deleting the "Default" profile is prevented; method returns false.</li> <li>- Renaming the "Default" profile is prevented; method returns false.</li> <li>- Appropriate messages are printed (e.g., "Cannot delete the default profile.").</li> <li>- The "Default" profile remains intact in the database.</li> <li>- No errors or exceptions occur during the attempts.</li> </ul>
Priority	Medium
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"> <li>- Deletion and renaming of "Default" profile are prevented.</li> <li>- "Default" profile remains unchanged.</li> <li>- Methods return false as expected.</li> </ul> <p>Fail if:</p> <ul style="list-style-type: none"> <li>- "Default" profile is deleted or renamed.</li> <li>- Methods incorrectly return true.</li> <li>- Errors occur during the process.</li> </ul>

**1.8 Basic UI Unit Test Items**

ID	UT65
Item to Test	Generated Tab Navigation Link
Test Description	Verify that tapping on the Generated tab link navigates to the Generated screen correctly.
Pre-Conditions	- The application is launched and is currently on the Main screen.
Test Steps	1. Locate the navigation tab bar at the bottom of the screen. 2. Tap on the Generated tab link. 3. Observe the screen after tapping the Generated tab.
Expected Results	- The application navigates to the Generated screen without errors. - A search bar with the text "Search Audios" is displayed near the top of the screen. - The Generated tab link button is highlighted in blue in the navigation tab bar.
Priority	High
Pass/Fail Criteria	Pass if: - The application navigates to the Generated screen successfully.  Fail if: - Navigation fails. - Errors or crashes occur.

ID	UT66
Item to Test	Documents Tab Navigation Link
Test Description	Verify that tapping on the Documents tab link navigates to the Documents screen correctly.
Pre-Conditions	- The application is launched and is currently on the Main screen.
Test Steps	1. Locate the navigation tab bar at the bottom of the screen. 2. Tap on the Documents tab link. 3. Observe the screen after tapping the Documents tab.
Expected Results	- The application navigates to the Documents screen without errors. - A search bar with the text "Search PDFs" is displayed near the top of the screen. - The Documents tab link button is highlighted in blue in the navigation tab bar.
Priority	High
Pass/Fail Criteria	Pass if: - The application navigates to the Documents screen successfully.  Fail if: - Navigation fails. - Errors or crashes occur.

ID	UT67
Item to Test	Text History Tab Navigation Link
Test Description	Verify that tapping on the Text History tab link navigates to the Text History screen correctly.
Pre-Conditions	- The application is launched and is currently on the Main screen.
Test Steps	1. Locate the navigation tab bar at the bottom of the screen. 2. Tap on the Text History tab link. 3. Observe the screen after tapping the Text History tab.
Expected Results	- The application navigates to the Text History screen without errors. - The text "Text History" is displayed near the top of the screen. - The Text History tab link button is highlighted in blue in the navigation tab bar.
Priority	High
Pass/Fail Criteria	Pass if: - The application navigates to the Text History screen successfully.  Fail if: - Navigation fails. - Errors or crashes occur.

ID	UT68
Item to Test	Settings Tab Navigation Link
Test Description	Verify that tapping on the Settings tab link navigates to the Settings screen correctly.
Pre-Conditions	- The application is launched and is currently on the Main screen.
Test Steps	1. Locate the navigation tab bar at the bottom of the screen. 2. Tap on the Settings tab link. 3. Observe the screen after tapping the Settings tab.
Expected Results	- The application navigates to the Settings screen without errors. - The text "Settings" is displayed near the top of the screen. - The Settings tab link button is highlighted in blue in the navigation tab bar.
Priority	High
Pass/Fail Criteria	Pass if: - The application navigates to the Settings screen successfully.  Fail if: - Navigation fails. - Errors or crashes occur.



ID	UT69
Item to Test	Main Tab Navigation Link
Test Description	Verify that tapping on the Main tab link navigates to the Main screen correctly.
Pre-Conditions	- The application is launched and is currently on the Settings screen.
Test Steps	1. Locate the navigation tab bar at the bottom of the screen. 2. Tap on the Main tab link. 3. Observe the screen after tapping the Main tab.
Expected Results	- The application navigates to the Main screen without errors. - The application icon and the text "EchoText" are displayed near the top of the screen. - The Main tab link button is highlighted in blue in the navigation tab bar.
Priority	High
Pass/Fail Criteria	Pass if: - The application navigates to the Main screen successfully.  Fail if: - Navigation fails. - Errors or crashes occur.

ID	UT70
Item to Test	Support Navigation Link in Settings
Test Description	Verify that tapping on the Support link navigates to the Support page correctly.
Pre-Conditions	- The application is launched and is currently on the Settings screen.
Test Steps	1. On the Settings screen, locate the Support navigation link. 2. Tap on the Support link. 3. Observe the screen after tapping the Support link.
Expected Results	- The application navigates to the Support page without errors. - The text "Support" is displayed near the top of the screen. - Relevant support information or options are displayed to the user.
Priority	Medium
Pass/Fail Criteria	Pass if: - The application navigates to the Support page successfully.  Fail if: - Navigation fails. - Errors or crashes occur.

## 2. Integration Testing

### 2.1 Text-to-Speech Conversion

ID	IT01
Item to Test	Text-to-Speech Conversion
Test Description	Test the process of converting input text to speech, verifying that text input is processed correctly, TTS engine generates audio, live playback works during conversion.
Pre-Conditions	<ol style="list-style-type: none"> <li>1. EchoText app is installed and launched</li> <li>2. Device has sufficient storage space</li> <li>3. Default voice profile is selected</li> </ol>
Test Steps	<ol style="list-style-type: none"> <li>1. Navigate to the main input screen</li> <li>2. Enter the following text in the input field: "Hello, this is a test of the text-to-speech system."</li> <li>3. Click the "Generate" button</li> <li>4. listen to make sure live playback is playing</li> <li>5. Wait for the conversion to complete</li> </ol>
Expected Results	<ol style="list-style-type: none"> <li>1. Text is successfully processed without errors</li> <li>2. Live playback started while text is being converted</li> <li>3. Audio playback produces clear, understandable speech</li> </ol>
Priority	High
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"> <li>- Text input is successfully captured without character loss or corruption</li> <li>- Generated audio file plays with clear, comprehensible speech quality</li> <li>- Generated audio matches the input text content accurately</li> <li>- Processing completes within 30 seconds for standard text input</li> <li>- Audio playback controls respond without lag or errors</li> </ul> <p>Fail if:</p> <ul style="list-style-type: none"> <li>- Text input is corrupted or lost during processing</li> <li>- Generated speech is incomprehensible or significantly distorted</li> <li>- Generated audio content does not match input text</li> <li>- Processing time exceeds 30 seconds without progress indication</li> <li>- Application crashes or freezes during any step</li> </ul>

## 2.2 Document Conversion

ID	IT02
Item to Test	Document Conversion
Test Description	Tests the conversion of a document to speech. Document parsing extracts text, text processor handles the content, TTS engine generates audio, live playback works during conversion.
Pre-Conditions	<ol style="list-style-type: none"> <li>1. EchoText app is installed and launched</li> <li>2. Device has sufficient storage space</li> <li>3. Default voice profile is selected</li> <li>4. Test1.pdf document is available in document view</li> </ol>
Test Steps	<ol style="list-style-type: none"> <li>1. Navigate to the “Documents” screen</li> <li>2. Press and hold on the listed document until a menu pops up</li> <li>3. press on “Generate” button in the menu</li> <li>4. Navigate to the “main” screen</li> <li>5. Ensure there is text pasted base on the document contents and the TTS conversion has started</li> <li>6. Ensure live playback has started playing</li> <li>7. Wait for the conversion to complete</li> </ol>
Expected Results	<ol style="list-style-type: none"> <li>1. Document content is correctly extracted</li> <li>2. Text is successfully converted to speech</li> <li>3. Live playback started while text is being converted</li> <li>4. Audio playback produces clear, understandable speech</li> </ol>
Priority	High
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"> <li>- Document text input is successfully captured without character loss or corruption</li> <li>- Generated audio file plays with clear, comprehensible speech quality</li> <li>- Generated audio matches the document text content accurately</li> <li>- Processing completes within 30 seconds for standard text input</li> <li>- Audio playback controls respond without lag or errors</li> </ul> <p>Fail if:</p> <ul style="list-style-type: none"> <li>- Document input is corrupted or lost during processing</li> <li>- Generated speech is incomprehensible or significantly distorted</li> <li>- Generated audio content does not match document text</li> <li>- Processing time exceeds 30 seconds without progress indication</li> <li>- Application crashes or freezes during any step</li> </ul>

## 2.3 Media Control

ID	IT03
Item to Test	Media Control System
Test Description	Verify the integration between the audio playback system and the user interface controls. Test includes play, pause, seek, and volume control functionality.
Pre-Conditions	<ol style="list-style-type: none"> <li>1. EchoText app is installed and launched</li> <li>2. At least two audio files have been generated and saved</li> <li>3. Device has working audio output</li> <li>4. No other audio is currently playing on device</li> </ol>
Test Steps	<ol style="list-style-type: none"> <li>1. Navigate to Generated Audio screen</li> <li>2. Select first audio file</li> <li>3. Press Play button and wait 5 seconds</li> <li>4. Press Pause button</li> <li>5. Move the seek bar to 50% position</li> <li>6. Press Play button again</li> <li>7. Adjust volume using device controls</li> <li>8. Press Forward button to skip to next audio</li> <li>9. Press Previous button to return to first audio</li> <li>10. Press Stop button</li> <li>11. Minimize app for 10 seconds</li> <li>12. Return to app</li> <li>13. Press Play button</li> <li>14. Lock device screen</li> </ol>
Expected Results	<ol style="list-style-type: none"> <li>1. Audio playback starts immediately when Play is pressed</li> <li>2. Audio pauses immediately when Pause is pressed</li> <li>3. Playback continues from 50% position after seeking</li> <li>4. Volume changes are reflected in playback</li> <li>5. Forward/Previous buttons correctly switch between audio files</li> <li>6. Stop button completely ends playback</li> <li>7. Playback position is retained when app is minimized</li> <li>8. Playback continues after app is reopened</li> </ol>
Priority	High
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"> <li>- All playback controls (Play, Pause, Stop) respond within 0.5 seconds</li> <li>- Seek bar accurately reflects and controls playback position</li> <li>- Audio transitions smoothly between paused and playing states</li> <li>- Audio quality remains consistent during all operations</li> <li>- Forward/Previous buttons correctly navigate between files</li> <li>- Volume controls work seamlessly with system audio</li> <li>- Playback state and position are preserved during app minimize/resume</li> </ul>

	<ul style="list-style-type: none"><li>- No audio glitches or stuttering during any operation</li></ul> <p>Fail if:</p> <ul style="list-style-type: none"><li>- Any control takes longer than 0.5 seconds to respond</li><li>- Seek bar position doesn't match actual playback position</li><li>- Audio stutters or glitches during state transitions</li><li>- Forward/Previous buttons skip incorrectly or fail to respond</li><li>- Volume controls affect wrong audio stream</li><li>- Playback state is lost when app is minimized</li><li>- Lock screen shows incorrect track information</li><li>- Lock screen controls are unresponsive</li><li>- Application crashes during any media operation</li><li>- Audio continues playing after Stop is pressed</li><li>- Multiple audio streams play simultaneously</li></ul>
--	--

## 2.4 Voice Profile Creation

ID	IT04
Item to Test	Voice Profile Creation
Test Description	Verify that new voice profiles can be created with custom settings and are properly stored in the system.
Pre-Conditions	<ol style="list-style-type: none"> <li>1. EchoText app is installed and launched</li> <li>2. Default voice profile exists in the system</li> <li>3. No custom profiles have been created</li> </ol>
Test Steps	<ol style="list-style-type: none"> <li>1. Navigate to Voice Profile section in Settings</li> <li>2. Press "Create New Profile" button</li> <li>3. Enter profile information: <ul style="list-style-type: none"> <li>- Name: "Custom Voice 1"</li> <li>- Model: "Kristin"</li> <li>- Speed: 1.2x</li> <li>- Pitch: +2</li> </ul> </li> <li>4. Press Save button</li> <li>5. Create second profile: <ul style="list-style-type: none"> <li>- Name: "Custom Voice 2"</li> <li>- Model: "Arctic"</li> <li>- Speed: 0.8x</li> <li>- Pitch: -1.0</li> </ul> </li> <li>6. Press Save button</li> </ol>
Expected Results	<ul style="list-style-type: none"> <li>- Both profiles are created with specified settings</li> <li>- Each profile appears in profile selection list</li> </ul>
Priority	High
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"> <li>- New profiles are created with all settings preserved</li> <li>- All UI elements update within 0.5 seconds</li> </ul> <p>Fail if:</p> <ul style="list-style-type: none"> <li>- Profile creation fails or creates with incorrect settings</li> <li>- UI takes longer than 0.5 seconds to update</li> </ul>

## 2.5 Passcode Creation

ID	IT05
Item to Test	Passcode Creation
Test Description	Verify the passcode creation process, including validation of matching entries and recovery code generation.
Pre-Conditions	<ul style="list-style-type: none"> <li>- EchoText app is installed and launched</li> <li>- No existing passcode is set</li> </ul>
Test Steps	<ol style="list-style-type: none"> <li>1. Navigate to Settings screen</li> <li>2. Press "Create Passcode" button</li> <li>3. Enter passcode "123456" on the first text field</li> <li>4. Enter "12345" on the second text field</li> <li>5. press enter</li> <li>6. Clear second text field</li> <li>7. Enter "123456" on the second text field</li> <li>8. press enter</li> <li>9. Observe recovery code dialog</li> <li>10. Note down 6-digit recovery code</li> <li>11. Press OK to close dialog</li> </ol>
Expected Results	<ul style="list-style-type: none"> <li>- Passcode creation process completes successfully</li> <li>- Recovery code is generated and displayed only once</li> <li>- Created passcode is required for subsequent app access</li> <li>- Recovery code is stored securely in the database</li> <li>- UI updates to show passcode protection is active</li> </ul>
Priority	Low
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"> <li>- Passcode creation requires exact matching entries</li> <li>- System generates unique 6-digit recovery code</li> <li>- Recovery code is displayed only once at creation</li> <li>- UI updates correctly to show protected state</li> </ul> <p>Fail if:</p> <ul style="list-style-type: none"> <li>- System accepts mismatched passcode entries</li> <li>- Recovery code is not generated</li> <li>- Recovery code is shown multiple times</li> <li>- UI fails to update protection state</li> <li>- App crashes during passcode creation</li> </ul>



## 2.6 Passcode Validation

ID	IT06
Item to Test	Passcode Validation
Test Description	Test the passcode validation system when accessing the app after minimize or close
Pre-Conditions	<ul style="list-style-type: none"> <li>- EchoText app is installed and launched</li> <li>- Passcode "123456" is set and active</li> </ul>
Test Steps	<ol style="list-style-type: none"> <li>1. Minimize app</li> <li>2. Return to app</li> <li>3. Enter incorrect passcode "111111"</li> <li>4. Observe error message</li> <li>5. Enter correct passcode "123456"</li> <li>6. Verify app unlocks</li> <li>7. Close app completely</li> <li>8. Relaunch app</li> <li>9. Enter correct passcode "123456"</li> <li>10. Verify app unlocks</li> </ol>
Expected Results	<ul style="list-style-type: none"> <li>- App requires passcode after minimize</li> <li>- App requires passcode after full restart</li> <li>- Incorrect passcode shows clear error message</li> <li>- Correct passcode grants access</li> <li>- Lock state persists across app restarts</li> <li>- UI accurately reflects locked/unlocked state</li> </ul>
Priority	Low
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"> <li>- App locks when minimized or closed</li> <li>- Correct passcode unlocks app</li> <li>- Incorrect passcode shows error message</li> <li>- Lock state persists through app restarts</li> <li>- UI accurately shows locked/unlocked state</li> <li>- Error messages are clear and helpful</li> </ul> <p>Fail if:</p> <ul style="list-style-type: none"> <li>- App fails to lock on minimize/close</li> <li>- Correct passcode fails to unlock app</li> <li>- No error message for wrong passcode</li> <li>- Lock state lost after restart</li> <li>- UI shows incorrect lock state</li> <li>- Error messages are unclear or missing</li> <li>- Lock screen can be bypassed</li> <li>- App crashes during validation</li> </ul>

## 2.7 Recovery Code Functionality

ID	IT07
Item to Test	Recovery Code Functionality
Test Description	Validate the recovery code system for regaining access when passcode is forgotten.
Pre-Conditions	<ul style="list-style-type: none"> <li>- EchoText app is installed and launched</li> <li>- Passcode is set and active</li> <li>- Recovery code is known</li> </ul>
Test Steps	<ol style="list-style-type: none"> <li>1. Minimize app</li> <li>2. Return to app</li> <li>3. Press "Forgot Passcode" button</li> <li>4. Enter incorrect recovery code</li> <li>5. Observe error message</li> <li>6. Enter correct recovery code</li> <li>7. Verify app unlocks</li> <li>8. Verify passcode is removed</li> </ol>
Expected Results	<ul style="list-style-type: none"> <li>- Recovery code system is accessible from lock screen</li> <li>- Incorrect recovery code shows error message</li> <li>- Correct recovery code grants access</li> <li>- Passcode protection is removed after recovery</li> <li>- System ready for new passcode creation</li> </ul>
Priority	Low
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"> <li>- Recovery code interface is accessible</li> <li>- Incorrect code shows error message</li> <li>- Correct recovery code grants access</li> <li>- Recovery process removes passcode protection</li> <li>- System is ready for new passcode creation</li> <li>- All related UI elements update correctly</li> </ul> <p>Fail if:</p> <ul style="list-style-type: none"> <li>- Recovery interface is inaccessible</li> <li>- Recovery code fails to unlock app</li> <li>- Recovery process doesn't remove passcode</li> <li>- System left in invalid state after recovery</li> <li>- UI fails to update after recovery</li> <li>- Multiple recovery codes work for same passcode</li> <li>- Recovery process crashes app</li> </ul>

## 2.8 Passcode Removal

ID	IT08
Item to Test	Passcode Removal
Test Description	Test the secure removal of passcode protection through settings.
Pre-Conditions	<ul style="list-style-type: none"> <li>- EchoText app is installed and launched</li> <li>- Passcode "789012" is set and active</li> </ul>
Test Steps	<ol style="list-style-type: none"> <li>1. Navigate to Settings</li> <li>2. Press "Remove Passcode" button</li> <li>3. Enter incorrect passcode "111111"</li> <li>4. Observe error message</li> <li>5. Enter correct passcode "789012"</li> <li>6. Verify passcode is removed</li> </ol>
Expected Results	<ul style="list-style-type: none"> <li>- Passcode removal requires current passcode verification</li> <li>- Incorrect passcode shows error message</li> <li>- Correct passcode allows removal</li> <li>- All passcode data is securely deleted</li> <li>- UI updates to reflect removed protection</li> <li>- Database updated to remove passcode data</li> <li>- App no longer requires passcode on minimize/close</li> </ul>
Priority	Low
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"> <li>- Removal requires current passcode verification</li> <li>- Incorrect passcode shows error message</li> <li>- Correct passcode allows removal</li> <li>- All passcode data is securely deleted</li> <li>- UI updates to reflect removed protection</li> <li>- App functions normally after removal</li> </ul> <p>Fail if:</p> <ul style="list-style-type: none"> <li>- Removal possible without verification</li> <li>- No error message for wrong passcode</li> <li>- Passcode data remains after removal</li> <li>- UI shows incorrect protection state</li> <li>- App crashes during removal</li> <li>- Protection state persists after removal</li> <li>- System allows bypass of verification</li> </ul>

## 2.9 Audio Search Functionality

ID	IT09
Item to Test	Audio Search Functionality
Test Description	Tests search feature in generated view. UI captures search input, search processor handles query, database executes search, and results display in UI.
Pre-Conditions	<ol style="list-style-type: none"> <li>1. EchoText app is installed and launched</li> <li>2. At least five audio files are generated with different names (sample, test, crazy, new news, testing)</li> </ol>
Test Steps	<ol style="list-style-type: none"> <li>1. Navigate to Generated Audio screen</li> <li>2. Type "test" in search bar</li> <li>3. Observe filtered results</li> <li>4. Clear search bar</li> <li>5. Type "sample" in search bar</li> <li>6. Observe filtered results</li> <li>7. Type "working" in search bar</li> <li>8. Observe filtered results</li> </ol>
Expected Results	<ol style="list-style-type: none"> <li>1. Search results appear as user types</li> <li>2. Results filter accurately based on input</li> <li>3. "No results" message shows when appropriate</li> <li>4. Search works across all file types</li> <li>5. Results update dynamically with system changes</li> <li>6. Search state preserves during app minimize</li> </ol>
Priority	Medium
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"> <li>- Search results appear within 0.5 seconds of typing</li> <li>- Partial matches are included in results</li> <li>- "No results" message displays when no matches found</li> <li>- UI remains responsive during search operations</li> </ul> <p>Fail if:</p> <ul style="list-style-type: none"> <li>- Search results take longer than 0.5 seconds to appear</li> <li>- Search fails to find existing matches</li> <li>- "No results" message fails to display</li> <li>- UI becomes unresponsive during search</li> <li>- Search function crashes the application</li> <li>- Results display incorrect file information</li> </ul>

## 2.10 Document Search Functionality

ID	IT10
Item to Test	Document Search Functionality
Test Description	Tests search feature in Document view. UI captures search input, search processor handles query, database executes search, and results display in UI.
Pre-Conditions	<ol style="list-style-type: none"> <li>1. EchoText app is installed and launched</li> <li>2. At least five PDF files are available with different names (sample.pdf, test.pdf, crazy.pdf, new news.pdf, testing.pdf)</li> </ol>
Test Steps	<ol style="list-style-type: none"> <li>1. Navigate to Generated Audio screen</li> <li>2. Type "test" in search bar</li> <li>3. Observe filtered results</li> <li>4. Clear search bar</li> <li>5. Type "sample" in search bar</li> <li>6. Observe filtered results</li> <li>7. Type "working" in search bar</li> <li>8. Observe filtered results</li> </ol>
Expected Results	<ol style="list-style-type: none"> <li>1. Search results appear as user types</li> <li>2. Results filter accurately based on input</li> <li>3. "No results" message shows when appropriate</li> <li>4. Search works across all file types</li> <li>5. Results update dynamically with system changes</li> <li>6. Search state preserves during app minimize</li> </ol>
Priority	Medium
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"> <li>- Search results appear within 0.5 seconds of typing</li> <li>- Partial matches are included in results</li> <li>- "No results" message displays when no matches found</li> <li>- UI remains responsive during search operations</li> </ul> <p>Fail if:</p> <ul style="list-style-type: none"> <li>- Search results take longer than 0.5 seconds to appear</li> <li>- Search fails to find existing matches</li> <li>- "No results" message fails to display</li> <li>- UI becomes unresponsive during search</li> <li>- Search function crashes the application</li> <li>- Results display incorrect file information</li> </ul>

## 2.11 Document Import Process

ID	IT11
Item to Test	Document Import
Test Description	Test the document import functionality by importing multiple PDFs and verifying they are correctly processed and stored with accurate metadata.
Pre-Conditions	<ol style="list-style-type: none"> <li>1. EchoText app is installed and launched</li> <li>2. At least 2 PDF files are available with different names (test1.pdf, test2.pdf)</li> </ol>
Test Steps	<ol style="list-style-type: none"> <li>1. Navigate to Documents screen</li> <li>2. Press Import button</li> <li>3. Select test1.pdf file from device storage</li> <li>4. Observe import process</li> <li>5. Repeat for test2.pdf</li> <li>6. Verify all imported PDFs are listed</li> <li>7. Check file metadata (date, size) is correct</li> </ol>
Expected Results	<ol style="list-style-type: none"> <li>1. All PDFs import successfully</li> <li>2. Files appear in correct app sections</li> <li>3. File metadata is accurate</li> <li>4. Import operations complete within 3 seconds</li> </ol>
Priority	High
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"> <li>- All files import successfully</li> <li>- Files appear in correct locations</li> <li>- Metadata is accurate and complete</li> <li>- Import completes within time limit</li> </ul> <p>Fail if:</p> <ul style="list-style-type: none"> <li>- Import fails for any supported format</li> <li>- Files appear in wrong sections</li> <li>- Metadata is incorrect or missing</li> <li>- Import exceeds time limit</li> <li>- App crashes during import</li> </ul>

## 2.12 File Rename and Delete

ID	IT12
Item to Test	File Rename and Delete
Test Description	Validate individual file operations including renaming files and deleting single files, ensuring database consistency and proper file system updates.
Pre-Conditions	<ol style="list-style-type: none"> <li>1. EchoText app is installed and launched</li> <li>2. At least 2 PDF files are available and imported with different names (test1.pdf, test2.pdf)</li> </ol>
Test Steps	<ol style="list-style-type: none"> <li>1. Navigate to Generated Audio screen</li> <li>2. press and hold on test1.pdf</li> <li>3. Press rename button</li> <li>4. Enter new name "Test Audio 1"</li> <li>5. press and hold test2.pdf</li> <li>6. press delete</li> <li>7. Confirm deletion</li> </ol>
Expected Results	<ol style="list-style-type: none"> <li>1. Rename operation completes successfully</li> <li>2. Delete operation removes file completely</li> </ol>
Priority	High
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"> <li>- Rename completes without errors</li> <li>- Delete removes file</li> <li>- Operations complete within 3 seconds</li> </ul> <p>Fail if:</p> <ul style="list-style-type: none"> <li>- Rename fails or corrupts files</li> <li>- Delete leaves residual data</li> <li>- Operations exceed 3 seconds</li> </ul>

**2.13 Batch File Deletion**

ID	IT13
Item to Test	Batch File Deletion
Test Description	Test the multi-select functionality and batch deletion capability, ensuring multiple files can be selected and deleted simultaneously.
Pre-Conditions	<ol style="list-style-type: none"> <li>1. EchoText app is installed and launched</li> <li>2. At least five audio files exist in the system</li> </ol>
Test Steps	<ol style="list-style-type: none"> <li>1. Navigate to Generated Audio screen</li> <li>2. Press Select button in top left</li> <li>3. Select three audio files</li> <li>4. Press Delete icon button at the bottom</li> <li>5. Confirm deletion in dialog</li> <li>6. Verify selected files are removed</li> <li>7. Verify remaining files are unaffected</li> </ol>
Expected Results	<ol style="list-style-type: none"> <li>1. Selection mode works correctly</li> <li>2. Multiple files delete successfully</li> <li>3. Unselected files remain intact</li> </ol>
Priority	High
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"> <li>- Select mode activates properly</li> <li>- Batch deletion completes successfully</li> <li>- Unselected files unaffected</li> </ul> <p>Fail if:</p> <ul style="list-style-type: none"> <li>- Select mode fails to activate</li> <li>- Batch deletion partially completes</li> <li>- Unselected files affected</li> </ul>



**2.14 File Sharing Functionality**

ID	IT14
Item to Test	File Sharing Functionality
Test Description	Verify the file sharing capabilities through iOS share sheet, ensuring files can be shared through different methods while maintaining file integrity and playback functionality.
Pre-Conditions	<ol style="list-style-type: none"> <li>1. EchoText app is installed and launched</li> <li>2. At least two audio files exist in the system</li> <li>3. Device has messaging capability configured</li> </ol>
Test Steps	<ol style="list-style-type: none"> <li>1. Navigate to Generated Audio screen</li> <li>2. Long press on an audio file</li> <li>3. press share button</li> <li>3. Verify iOS share sheet appears</li> <li>4. Share file via Messages</li> <li>5. Return to Generated Audio screen</li> <li>6. Long press different audio file</li> <li>7. press share button</li> <li>8. Share via iMessage</li> <li>9. Verify shared files are received and playable</li> </ol>
Expected Results	<ol style="list-style-type: none"> <li>1. Share sheet appears properly</li> <li>2. Files share successfully</li> <li>3. Shared files remain playable</li> <li>4. Original files unaffected</li> </ol>
Priority	High
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"> <li>- Share sheet appears correctly</li> <li>- Files share without corruption</li> <li>- Shared files play properly</li> <li>- Original files remain intact</li> </ul> <p>Fail if:</p> <ul style="list-style-type: none"> <li>- Share sheet fails to appear</li> <li>- Files corrupt during sharing</li> <li>- Shared files won't play</li> <li>- Original files affected</li> </ul>

## 2.15 Progress Indication System

ID	IT15
Item to Test	Progress Indication System
Test Description	Verify that progress indicators accurately reflect the status of background operations such as TTS processing and batch processing of documents.
Pre-Conditions	<ol style="list-style-type: none"> <li>1. EchoText app is installed and launched</li> <li>2. At least three PDF documents imported</li> <li>3. User is on main screen</li> <li>4. No conversion process is running</li> </ol>
Test Steps	<ol style="list-style-type: none"> <li>1. Single File Progress Test: <ul style="list-style-type: none"> <li>• Enter text "This is a test of the progress system"</li> <li>• Press Convert button</li> <li>• Observe bottom progress bar during conversion</li> <li>• Wait for completion</li> </ul> </li> <li>2. Short Text Progress Test: <ul style="list-style-type: none"> <li>• Enter single word "Test"</li> <li>• Press Convert button</li> <li>• Observe progress bar behavior</li> </ul> </li> <li>3. Long Text Progress Test: <ul style="list-style-type: none"> <li>• Paste a paragraph with 500+ words</li> <li>• Press Convert button</li> <li>• Observe progress bar behavior</li> </ul> </li> <li>4. Batch Processing Progress Test: <ul style="list-style-type: none"> <li>• Navigate to Documents screen</li> <li>• Select three PDFs for batch conversion</li> <li>• Start batch process</li> <li>• Observe top right counter (1/3, 2/3, 3/3)</li> <li>• Observe bottom progress bar for current PDF</li> <li>• Wait for each file to complete</li> </ul> </li> <li>5. Progress Interruption Test: <ul style="list-style-type: none"> <li>• Start batch conversion of three PDFs</li> <li>• While second PDF is processing, press Terminate</li> <li>• Observe progress indicators</li> </ul> </li> <li>6. App State Test: <ul style="list-style-type: none"> <li>• Start batch conversion</li> <li>• Minimize app during conversion</li> <li>• Return to app</li> <li>• Observe progress indicators</li> </ul> </li> </ol>
Expected Results	<ol style="list-style-type: none"> <li>1. Bottom progress bar shows conversion progress smoothly</li> <li>2. Batch counter displays current/total files accurately</li> <li>3. Progress indicators update in real-time</li> </ol>

	4. Progress state preserved during app minimize 5. Progress resets properly after completion/termination
Priority	Medium
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"> <li>- Single file progress bar updates smoothly from 0% to 100%</li> <li>- Batch counter format displays correctly (e.g., "2/3")</li> <li>- Batch counter updates immediately when file completes</li> <li>- Progress bar resets between each file in batch</li> <li>- Progress indicators remain accurate after app minimize/resume</li> <li>- Termination properly resets all progress indicators</li> <li>- UI remains responsive while progress is updating</li> <li>- Progress accurately reflects actual conversion status</li> <li>- Long text conversions show continuous progress updates</li> </ul> <p>Fail if:</p> <ul style="list-style-type: none"> <li>- Batch counter shows incorrect numbers</li> <li>- Progress bar fails to reset between files</li> <li>- Progress state lost during app minimize</li> <li>- Progress indicators show incorrect status</li> <li>- Termination leaves progress indicators in incorrect state</li> <li>- UI becomes unresponsive during progress updates</li> <li>- Progress bar completes before actual conversion</li> <li>- Progress indicators missing for any conversion type</li> <li>- Batch counter fails to update after file completion</li> <li>- System crashes during progress updates</li> <li>- Progress bar or counter disappears unexpectedly</li> <li>- Multiple progress bars appear simultaneously</li> </ul>

### 3. System Testing

#### 3.1 Core Application Functionalities

ID	ST1
Item to Test	Core Application Functionalities
Test Description	Validate the core functionalities of the EchoText application, focusing on text-to-speech (TTS) generation. Ensure users can input text (manual, copy-paste, and document import), generate audio files, and listen to playback. Verify that generated audio is saved in the "Generated Audio" list for future access. Validate the ability to process batch document conversions. Ensure that playback controls (play, pause, rewind, fast-forward, and speed adjustment) operate correctly without errors.
Pre-Conditions	<ol style="list-style-type: none"> <li>1. EchoText app is installed and launched.</li> <li>2. test1.pdf and test2.pdf are imported and available in documents view.</li> </ol>
Test Steps	<p>Navigate to the main TTS input screen.</p> <ol style="list-style-type: none"> <li>1. Input Text on the Main Tab <ul style="list-style-type: none"> <li>• Type, Copy or Paste the following text in the white box on the Main Tab. <ul style="list-style-type: none"> <li>○ “What is TTS? TTS stands for text-to-speech, which is a technology that converts written text into spoken words. TTS uses artificial intelligence (AI) to create a voice that sounds human-like.</li> </ul> </li> </ul> </li> <li>2. Mute audio during generation <ul style="list-style-type: none"> <li>• Press the gray volume button on the bottom left to mute the audio.</li> </ul> </li> <li>3. Terminate <ul style="list-style-type: none"> <li>• While the audio is playing, wait for the progress bar to disappear.</li> <li>• Next, press the red stop button at the bottom to terminate the audio.</li> <li>• Click on the trash icon to clear text.</li> </ul> </li> <li>4. Playback via Generated Tab <ul style="list-style-type: none"> <li>• Go to Generated tab to view and play previously generated audio.</li> <li>• Click on the first audio.</li> <li>• Audio will play automatically.</li> <li>• Click on the center pause button to pause the audio.</li> <li>• Click on the forward and backward buttons to move the audio.</li> <li>• The audio speed is set to 1x; to adjust speed click on the 1x speed icon.</li> </ul> </li> </ol>

	<p>5. Play via imported PDF.</p> <ul style="list-style-type: none"> <li>• Navigate to Documents Tab at the bottom of the screen.</li> <li>• Select test1.pdf or test2.pdf and click on the document icon to the right of the pdf.</li> <li>• Now click Select Pages at the bottom of the screen.</li> <li>• Input range of pages or select the page numbers from the PDFs.</li> <li>• Now click generate the top right of the screen to generate audio.</li> </ul>
Expected Results	<p>1. System accepts input text correctly and displays it as entered.</p> <p>2. Audio playback mutes after pressing the mute button.</p> <p>3. Audio playback terminates upon pressing the delete button.</p> <p>4. TTS generates audio files accurately from input text or imported PDF.</p> <p>5. The system should play the selected audio file automatically without glitches.</p> <p>6. The Pause button should successfully stop the playback.</p> <p>7. The Forward and Backward buttons should allow precise movement through the audio, and the playback should resume correctly.</p> <p>8. The audio speed should adjust as expected when the user clicks on the 1x speed icon, with the selected speed taking effect immediately. The system should revert to 1x speed when clicked again.</p> <p>9. The PDF should be successfully selected and displayed in the system.</p> <p>10. The page range input or manual selection should work correctly, with the selected pages shown accurately.</p> <p>11. Clicking Generate should trigger the audio generation process, and the generated audio should be available in the Generated Tab once the process is complete.</p> <p>12. The generated audio should match the selected pages and play without any issues.</p>
Priority	High
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"> <li>- The system successfully accepts the input text, mutes and terminates the audio, and processes the PDF file correctly.</li> <li>- The playback functions (play, pause, forward, backward) and speed adjustment work as expected.</li> <li>- The generated audio matches the selected pages of the PDF and plays without issues.</li> </ul>

	<ul style="list-style-type: none"><li>- The system does not experience any glitches, errors, or unexpected behavior.</li></ul> <p>Fail if:</p> <ul style="list-style-type: none"><li>- The input text is not displayed correctly, with truncation or formatting issues.</li><li>- The mute button does not stop the audio, or the stop button does not terminate the playback properly.</li><li>- The playback features (pause, forward, backward) do not work correctly, causing playback to freeze, skip, or fail.</li><li>- The speed adjustment does not take effect or reverts unexpectedly.</li><li>- The PDF does not open correctly, or the page range selection is inaccurate.</li><li>- The audio generated from the PDF does not correspond to the selected pages or has errors during playback.</li></ul>
--	---

### 3.2 File and Queue Management

ID	ST2
Item to Test	File and Queue Management
Test Description	Ensure the proper functionality of file management operations, including importing files, renaming, deleting, and exporting generated audio files. Verify queue management operations such as adding, deleting, and reordering files. Validate file persistence and bookmarking functionality after the application restarts. Test file operations for large files (up to 100 MB).
Pre-Conditions	<ol style="list-style-type: none"> <li>1. EchoText app is installed and launched.</li> <li>2. Files and audio are already generated or imported.</li> <li>3. test1.pdf and test2.pdf are imported and available in documents view.</li> </ol>
Test Steps	<ol style="list-style-type: none"> <li>1. Rename an existing file and verify the change is saved. <ul style="list-style-type: none"> <li>• Navigate to Generated Tab.</li> <li>• Click on the first file in the queue.</li> <li>• Hold the file in the Generated Tab and select Rename from the pop-up menu to rename the file.</li> <li>• Enter “test1.pdf” to rename the file.</li> <li>• Exit the app or select Main Tab again.</li> <li>• Now return to Generated Tab and ensure the file is in the queue with the new name “test1.pdf”.</li> </ul> </li> <li>2. Delete a file and verify it is removed from the list. <ul style="list-style-type: none"> <li>• Navigate to Generated Tab.</li> <li>• Hold the file in the Generated Tab and select Delete from the pop-up menu to remove the file.</li> <li>• Now the delete confirmation box will pop up.</li> <li>• Press the delete button to finalize deletion of the file.</li> </ul> </li> <li>3. Add audio files to the playback queue. <ul style="list-style-type: none"> <li>• Navigate to Generated Tab.</li> <li>• Hold the file in the Generated Tab and select Add to Queue from the pop-up menu.</li> <li>• Click on the queue icon in the left corner of the audio player box.</li> <li>• Ensure that the selected file is in the queue to be played next.</li> </ul> </li> <li>4. Share the generated audio file. <ul style="list-style-type: none"> <li>• Navigate to Generated Tab.</li> <li>• Hold the file in the Generated Tab and select Share button from the pop-up menu.</li> <li>• Now share the file via messages, email, etc.</li> </ul> </li> <li>5. View imported document. <ul style="list-style-type: none"> <li>• Navigate to Generated Tab.</li> </ul> </li> </ol>

	<ul style="list-style-type: none"> <li>• Hold the file in the Generated Tab and select View Document button from the pop-up menu.</li> </ul> <p>6. Get more information about the file.</p> <ul style="list-style-type: none"> <li>• Navigate to Generated Tab.</li> <li>• Hold the file in the Generated Tab and select More Info button from the pop-up menu.</li> <li>• A new screen with the following information will pop up:             <ul style="list-style-type: none"> <li>○ Date &amp; Time</li> <li>○ Name</li> <li>○ Length</li> <li>○ Model</li> <li>○ Speed</li> <li>○ Pitch</li> <li>○ Document</li> <li>○ Favorite</li> </ul> </li> </ul> <p>7. Restart the app and verify file persistence and bookmarks.</p> <ul style="list-style-type: none"> <li>• Navigate to Generated Tab.</li> <li>• Ensure that the generated audio for above text appears in the tab.</li> <li>• Tap on the “test1.pdf” audio file.</li> <li>• The audio will automatically play.</li> <li>• Pause the audio at 10 seconds using the pause button in the center at the bottom.</li> <li>• Exit the app or select Main Tab again.</li> <li>• Now return to Generated Tab and ensure the file is present.</li> <li>• Select the “test1.pdf” and ensure it plays from 10 second time stamp.</li> </ul> <p>8. Mark “test1.pdf” file as favorite.</p> <ul style="list-style-type: none"> <li>• Navigate to generated tab.</li> <li>• Look for a heart outline next to “test1.pdf” file.</li> <li>• Click on the heart so it is filled and opaque.</li> </ul>
Expected Results	<p>1. The renamed file should appear correctly in the Generated Tab with the new name “test1.pdf”, reflecting the change made earlier.</p> <p>2. The Delete Confirmation Box should display correctly, with a clear option to confirm or cancel the deletion.</p> <p>3. Once the Delete button is pressed, the file should be removed from the Generated Tab and no longer appear in the list.</p> <p>4. The Add to Queue action should successfully add the selected file to the playback queue.</p> <p>5. The Queue Icon should show the updated list of files, including the newly added file.</p>



	<p>6. The selected file should be visible in the queue and marked to be played next.</p> <p>7. The system should maintain the correct order of the queue, and files should play in sequence from the queue when triggered.</p> <p>8. The user should be able to select their preferred sharing method (such as Messages or Email) and complete the process without errors.</p> <p>9. The View Document option should open the selected document in a viewer within the application.</p> <p>10. When the More Info option is selected, a new screen or modal should appear showing the following information about the selected file:</p> <ul style="list-style-type: none"> <li>• Date &amp; Time: Correct date and time the file was created or modified.</li> <li>• Name: The file name is displayed correctly.</li> <li>• Length: The length of the generated audio should be shown accurately.</li> <li>• Model: The correct text-to-speech model or engine used for the generation.</li> <li>• Speed: The audio speed used during generation should be displayed.</li> <li>• Pitch: The pitch setting used should be shown accurately.</li> <li>• Document: The name of the document from which the audio was generated should be shown.</li> <li>• Favorite: Whether the file is marked as a favorite should be indicated if applicable.</li> </ul> <p>11. The system should properly save and restore the paused position of the audio, ensuring playback continues from the correct point.</p> <p>12. The file should now be marked as a favorite in the Generated Tab, and the heart icon should remain filled and opaque until manually unchecked.</p>
Priority	High
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"> <li>- All actions (rename, delete, add to queue, share, view document, more info, bookmarking, favorite) work as expected without errors.</li> <li>- The file changes (e.g., rename, delete, mark as favorite) are reflected immediately and persist correctly.</li> <li>- The test1.pdf file behaves as expected across app sessions (e.g., retains name, bookmarks, favorite status).</li> </ul> <p>Fail if:</p>

	<ul style="list-style-type: none"><li>- Any action (rename, delete, add to queue, share, etc.) does not work as expected or causes an error.</li><li>- The file does not reflect changes (e.g., name, favorite status) after performing the action.</li><li>- The file is not preserved after restarting the app or navigating away from the Generated Tab.</li><li>- The file does not resume playback from the correct position or is not marked as a favorite when the heart icon is clicked.</li></ul>
--	--

### 3.3 User Settings and Customization

ID	ST3
Item to Test	User Settings and Customization
Test Description	Validate the functionality of user settings, including customizing voice profiles, adjusting TTS parameters like pitch and speed, and toggling between light and dark themes. Verify that settings changes persist across sessions and can be reset to defaults. Ensure security features like password protection work as expected. Confirm all settings sections and menus are accessible and navigable.
Pre-Conditions	<ol style="list-style-type: none"> <li>1. EchoText app is installed and launched.</li> <li>2. Default settings are functional.</li> </ol>
Test Steps	<ol style="list-style-type: none"> <li>1. Customize voice profiles (e.g., adjust pitch, speed). <ul style="list-style-type: none"> <li>• Navigate to the settings menu on the bottom right of the screen.</li> <li>• Select Voice Profile from the menu.</li> <li>• It will already be set to default.</li> <li>• Select the plus icon on the top right to add a new Voice Profile.</li> <li>• Type “Test Profile” in the Add New Profile pop up box to create a new profile.</li> <li>• Press back on the top left corner of the screen.</li> <li>• Press Select Model to choose arctic as your new model.</li> <li>• Press back on the top left corner of the screen.</li> <li>• Adjust Speed and Pitch as follows using the seek bar. <ul style="list-style-type: none"> <li>○ Speed: 1.5x</li> <li>○ Pitch: +2</li> </ul> </li> </ul> </li> <li>2. Toggle between light and dark themes. <ul style="list-style-type: none"> <li>• Navigate to the setting menu and select Theme and Display.</li> <li>• Turn on Override System Theme.</li> <li>• Choose Dark to change theme and display.</li> </ul> </li> <li>3. Enable password protection and verify functionality. <ul style="list-style-type: none"> <li>• Navigate to the settings menu and select Security.</li> <li>• Choose Add Password.</li> <li>• Type “123456” in New Passcode field.</li> <li>• Type “123456” in the Confirm Passcode field.</li> <li>• Select Save at the top right to save the password.</li> <li>• Recovery Code Box will pop up with your recovery code.</li> <li>• Save this number securely for later use and press okay.</li> </ul> </li> <li>4. Change Password <ul style="list-style-type: none"> <li>• Navigate to the setting menu and select Security.</li> <li>• Choose Change Password.</li> </ul> </li> </ol>

	<ul style="list-style-type: none"> <li>• Enter your current passcode “123456” in the Current Passcode field.</li> <li>• Enter “112233” in the New Passcode field.</li> <li>• Reenter “112233” in the Confirm Passcode field.</li> <li>• Select Save at the top right to save the password.</li> <li>• Recovery Code Box will pop up with your new recovery code.</li> <li>• Save this number securely for later use and press okay.</li> </ul> <p>5. Remove Password</p> <ul style="list-style-type: none"> <li>• Navigate to the setting menu and select Security.</li> <li>• Choose Remove Password.</li> <li>• Enter current passcode “112233” in the Current Passcode field.</li> <li>• Select Remove Password at the bottom of the screen to remove password.</li> </ul> <p>6. Reset settings to defaults and verify changes.</p> <ul style="list-style-type: none"> <li>• Navigate to the settings menu and select Advanced.</li> <li>• Press Rest Settings in Red to revert back to default settings.</li> </ul>
Expected Results	<ol style="list-style-type: none"> <li>1. The system should allow the creation of a new voice profile called “<b>Test Profile</b>” without errors.</li> <li>2. After selecting the <b>Arctic</b> voice model, the system should save and apply this model to the new profile.</li> <li>3. The <b>Speed</b> should be successfully adjusted to <b>1.5x</b> and the <b>Pitch</b> should be adjusted to <b>+2</b> using the seek bars.</li> <li>4. The <b>Test Profile</b> should now reflect these changes (Arctic model, Speed: 1.5x, Pitch: +2).</li> <li>5. The changes should persist even after exiting and returning to the <b>Voice Profile</b> settings.</li> <li>6. The <b>Override System Theme</b> toggle should be enabled without errors.</li> <li>7. After selecting <b>Dark</b>, the system should immediately switch to dark mode, and the theme should be reflected across the entire application.</li> <li>8. All UI elements (e.g., background, text, buttons) should change to the dark theme and remain legible with appropriate contrast.</li> <li>9. The <b>Add Password</b> option should work correctly, allowing the user to set a new password.</li> <li>10. The system should prompt the user with a <b>Recovery Code</b> after saving the password. The code should be displayed clearly and can be copied or written down for future use.</li> <li>11. The <b>Change Password</b> functionality should allow the user to change their passcode successfully.</li> <li>12. After saving the new password, the system should generate and display a new <b>Recovery Code</b>.</li> </ol>

	<p>13. The <b>Recovery Code</b> should be visible, and the user should be able to securely store it for later use.</p> <p>14. After selecting <b>Remove Password</b>, the system should successfully remove the password and display a confirmation message or prompt indicating the removal was successful.</p> <p>15. Once the password is removed, the user should no longer be prompted to enter the passcode to access restricted areas of the app (e.g., settings).</p> <p>16. No personalized settings (e.g., themes, passwords, voice profiles) should remain after the reset, and the application should function as it did when first installed, with all settings set to their default configurations.</p>
Priority	Medium
Pass/Fail	<p>Pass if:</p> <ul style="list-style-type: none"> <li>- The Test Profile is created correctly and applies the changes (Arctic model, Speed: 1.5x, Pitch: +2).</li> <li>- The Dark theme is applied correctly across all UI elements, and the Override System Theme toggle works.</li> <li>- The password functionality works as expected:</li> <li>- The user can set, change, and remove the password with proper Recovery Code prompts.</li> <li>- The password removal works correctly, and the user is no longer prompted for the passcode after removal.</li> <li>- After pressing Reset Settings, all settings are reset to defaults and no personalized settings remain.</li> </ul> <p>Fail if:</p> <ul style="list-style-type: none"> <li>- The Test Profile is not created or does not reflect the correct settings (Arctic model, Speed: 1.5x, Pitch: +2).</li> <li>- The Dark theme does not apply or causes UI issues (e.g., unreadable text, incorrect background color).</li> <li>- The password settings (Add, Change, Remove) do not work as expected, or the Recovery Code is not shown or does not function.</li> <li>- Remove Password fails, and the user is still prompted to enter a passcode after removal.</li> <li>- Reset Settings does not work, leaving personalized settings in place or the app does not return to its default state.</li> </ul>