VoiceMail is an innovative application that converts emails into speech, allowing users to read and send emails while also reading them to them using a chosen voice. The test plan aims to ensure the platform's seamless functionality and reliability. The project includes functions like email client sign-in, email sending and receiving, filtering emails based on keywords and local user login and sign-out. Users can select specific emails to be read aloud, choose from various voice options, request the reading of all unread emails, and schedule reading times. The project can manage multiple email addresses and logins, catering to diverse user needs. Following the thorough testing procedures in this plan, our aim is to make sure the VoiceMail platform works smoothly. It should be a dependable and efficient tool for managing email communications.

Google Sheet Reference: For detailed test case steps, expected results, and additional information, please refer to the following document: [https://docs.google.com/spreadsheets/d/14SxfSm6KBDVehYfWzhc-2Xbx00R-G5dya5EqAGN21IY/edit#gid=1119765760].

Scope:

This testing effort will cover the following aspects of the VoiceMail project:

1. **Cross-browser Compatibility:** Ensure smooth functionality across different web browsers.
2. **Sign In:** Validate sign-in processes, including successful and unsuccessful attempts, session persistence, and security measures.
3. **Inbox Page**: Confirm proper email display, selection, filtering, and scheduling functionalities.
4. **Sign Out:** Verify successful sign-out functionality and authentication status across different browsers.
5. **Reading Email Page:** Validate correct display and playback of email details, including attachments and non-English content.
6. **Compose Email Page:** Ensure successful email sending and attachment handling.
7. **Settings Page:** Verify the ability to change voice preferences and save settings.

Platform and Browser Coverage:

Testing will be conducted across popular web browsers like Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari, covering both desktop and mobile platforms.

Test Case:

The test case list has been provided with detailed test cases for every page of the VoiceMail project, encompassing each functionality. Specific steps to be executed, expected results, and any preconditions required for the test are explained in each test case. Screenshots of the comprehensive test cases are attached below (Figure 1) for reference and implementation during the testing phase.
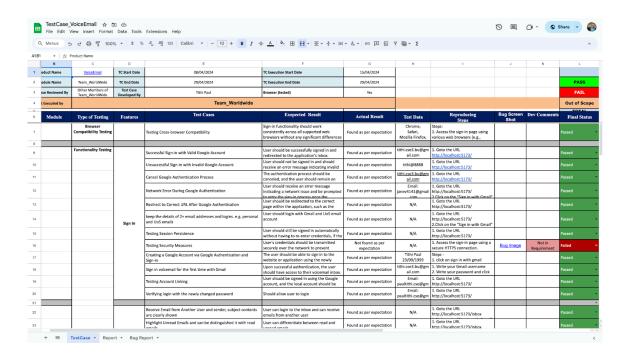
Figure 1: Test_Cases list for testing VoiceMail

Thorough testing of the VoiceMail platform has been ensured to guarantee its reliability and functionality across different browsers and scenarios. The system was also tested based on the primary requirement and functionalities not explicitly mentioned but deemed essential for basic voicemail functionality. As a result, most of the test cases passed, while some others failed. The whole test cases can be reviewed, along with how they were tested, the expected results, actual results, and test status.

Test Report:

In the test report for the project, there were a total of 78 test cases evaluated. Out of these, 65 test cases passed successfully while 13 test cases failed to meet the expected criteria.

This breakdown offers valuable insights into the performance of the system under test. The high number of passed test cases indicates that the majority of functionalities are working as intended and meeting the specified requirements. However, the presence of failed test cases highlights areas where the system may have defects or issues that need to be addressed.
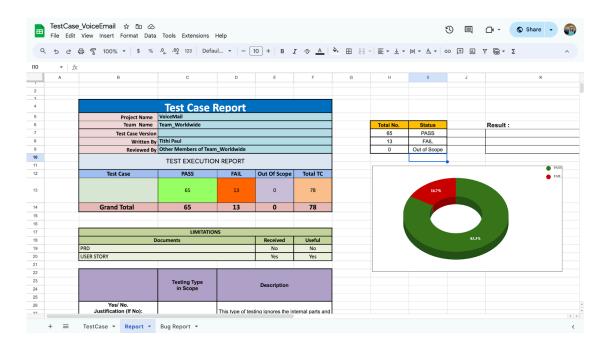
Figure 2: Test Report for testing VoiceMail

The ratio of passed to failed test cases provides a quantitative measure of the overall test coverage and effectiveness. A higher ratio of passed to failed test cases generally indicates a more stable and reliable system. Conversely, a lower ratio may suggest areas of concern that require further investigation and resolution.

Bug Report:

The bug report is carefully put together to include important details like the bug ID and the specific test case number that caused the issue. Each bug is described thoroughly so everyone understands what's going on. Moreover, the report breaks down step-by-step how to make the bug happen again. This helps everyone see exactly how to replicate the problem for further investigation.

Figure 3: Bug Report after testing VoiceMail

Additionally, the report points out exactly where in the VoiceMail system the bug popped up. This helps developers quickly find and fix the issue.

Also, if needed, the report includes screenshots showing what the bug looks like. These pictures help everyone visualise the problem and understand it better.

In short, the bug report helps keep track of and deal with any problems that come up during testing, ensuring the VoiceMail system works smoothly.

Test Environment Details

- **Hardware:**
  - MacBook Pro, Apple M2
- **Software:**
  - MacOS: Sonoma 14.4.1
  - Google Chrome [Version - 124.0.6367.92]

Test Approach

In the development cycle of the project, ensuring robust functionality and reliability is paramount. To achieve this, a comprehensive test approach is imperative, encompassing various methodologies, techniques, and meticulous documentation. This approach not only validates the system's performance but also aids in identifying and rectifying any anomalies efficiently. Here lies a detailed overview of our test approach, outlining the essential components for each test case.

1. **Type of Testing:** Each test case undergoes multiple types of testing to ensure thorough validation of the project.
2. **Features:** All project features undergo meticulous testing to validate functionality.
3. **Test Cases:** A comprehensive set of test cases is developed to cover every aspect of the project.
4. **Expected Result:** Clearly defined benchmarks against which system performance is evaluated.
5. **Actual Result:** Recorded and compared against expected outcomes to identify discrepancies.
6. **Test Data:** Diverse data sets simulate real-world usage scenarios for comprehensive testing.
7. **Reproducing Steps:** Documented steps provide a structured approach for executing test cases.
8. **Bug Screen Shot:** Visual evidence aids in understanding and resolving reported issues.
9. **Developer Comments:** Insights into root causes and potential solutions facilitate efficient issue resolution.

10. **Final Status:** Recorded to indicate test case pass or fail, providing an overview of system performance.

In summary, our test approach for the VoiceMail project is designed to ensure thorough validation of system functionality and reliability. Through the incorporation of diverse testing methodologies and meticulous documentation, a robust and dependable email management solution is aimed to be delivered, meeting the needs of our users.