Test Plan for AmaderHR Website

Adri saha

1. Introduction

The **Amamder HR Web App** is an advanced and efficient platform developed to streamline key human resource functions within an organization. This web application provides a comprehensive solution for managing employee data, banking details, educational qualifications, and address records while ensuring secure access and efficient system monitoring. The app focuses on enhancing HR operations by automating and simplifying processes related to employee management, payroll, recruitment, and more.

2. Scope

The **Amamder HR Web App** is designed to streamline specific HR functionalities, offering efficient tools for managing banking, education, and address records, along with secure access control. The scope of this application is as follows:

1. **Bank Management**
    - Maintaining employee banking details for secure and seamless salary transfers.
    - Ensuring compliance with banking regulations and confidentiality of financial data.
    - Providing tools for generating reports related to salary disbursement and account changes.
2. **Education Management**
    - Recording and managing employee educational qualifications and certifications.
    - Tracking training programs and professional development milestones.
    - Supporting skill assessment and career progression planning.
3. **Address Management**
    - Storing and managing employee address records for accurate documentation.
    - Enabling updates to address details for relocations or contact changes.
    - Ensuring the availability of address records for official correspondence and compliance.
4. **Login and Logout Management**
    - Implementing secure authentication protocols for employee system access.
    - Maintaining detailed logs of login and logout activities for monitoring and auditing purposes.
    - Enhancing system security with password policies and user role-based access control.

3. Test Objectives

- **Verify Functional Requirements**: Ensure that all specified functional requirements are accurately implemented, with each feature performing as expected across the application.
- **Usability and User Experience**: Assess the user interface to ensure the website is intuitive, easy to navigate, and provides a seamless experience for both employees and HR personnel.
- **Performance Testing**: Validate that the web application performs optimally under various load conditions, ensuring fast response times and minimal downtime during high traffic.
- **Security Assessment**: Identify and resolve any security vulnerabilities within the system, ensuring the protection of sensitive employee data and compliance with data privacy standards.
- **Payment Gateway Integration**: Confirm smooth and reliable integration with external payment gateways, ensuring accurate transaction processing and secure payment handling.
- **Cross-Browser Compatibility**: Test the web application on various browsers (e.g., Chrome, Firefox, Safari, Edge) to ensure consistent functionality and appearance across all platforms.
- **Cross-Device Compatibility**: Ensure that the web application is fully responsive and optimized for use on various devices, including desktops, tablets, and smartphones.
- **Error Handling and Validation**: Ensure that the app properly handles all errors and exceptions, providing clear error messages and preventing system crashes.
- **Data Integrity**: Validate that data entered into the system is correctly stored and retrieved, ensuring accuracy in employee records, banking details, and other sensitive information.
- **Backup and Recovery Testing**: Verify the functionality of backup systems and the ability to restore the application to its last stable state in case of a failure.
- **User Access Control**: Ensure that the role-based access control mechanisms are working as intended, granting proper permissions to HR personnel, employees, and admins.
- **Compliance Check**: Ensure that the application adheres to relevant local laws and industry standards, such as labor laws, data protection regulations, and accessibility standards.

4. Resources

Human Resources:

- Test Manager: John Doe
- Test Lead: Jane Smith
- Testers: 4 SQA engineers

Hardware Resources:

- Test servers for different environments (development, staging, production)
- Various devices for cross-device testing (desktops, tablets, smartphones)

Software Resources:

- Test management tools (e.g., JIRA, TestRail)
- Automation tools (Selenium, Appium)
- Performance testing tools (JMeter)

5. Schedule

Activity	**Start Date**	**End Date**
Test Planning	01-November-2024	05-November-2024
Test Design	06-November-2024	15-November-2024
Test Environment Setup	10-November-2024	12-November-2024
Test Case Development	13-November-2024	20-November-2024
Test Execution	21-November-2024	28-November-2024
Performance Testing	29-November-2024	30-November-2024
Performance Testing	01-December-2024	05-December-2024
UAT (User Acceptance Test)	5-December-2024	8-December-2024
Test Closure	16-December-2024	18-December-2024

6. Test Environment

1. **Development Environment**
    - **What it is**: This is where the app is being built. It's hosted on internal servers and used by the developers to write and test new code.
    - **What happens here**: Developers work on new features and fix bugs. The version of the app here is not final and may change frequently.
    - **Purpose**: To test small parts of the app and make sure new code works before moving it to the next stage.
2. **Staging Environment**
    - **What it is**: This environment is a copy of the real app, set up just like the live version. It's used for final testing before the app goes live.
    - **What happens here**: The app is almost finished and tested with realistic data. This is where the QA team checks if everything works well, including how it looks on different browsers and devices, and tests for performance and security.
    - **Purpose**: To make sure everything works as expected and there are no issues before launching the app to the public.
3. **Production Environment**
    - **What it is**: This is the live version of the app, where users will interact with it. It's hosted on real servers and accessible to everyone.
    - **What happens here**: After the app passes all tests in the staging environment, it's launched here. The app runs smoothly and is monitored for any issues that might affect users.
    - **Purpose**: To make sure the app works well for real users. We also check it for any last-minute bugs after it's released.

7. Test Deliverables

- **Test Plan**
    - **Description**: A comprehensive document that outlines the overall testing strategy, objectives, scope, approach, resources, and schedule for testing the **Amamder HR Web App**.
    - **Relevance**: The test plan will cover the testing of critical features like bank management, education management, address management, and login/logout functionality. It will define the testing methods, testing phases, and allocate tasks to the QA team to ensure smooth and efficient testing of the web app across all environments.

- **Test Cases**
    - **Description**: Detailed descriptions of individual test scenarios, specifying the conditions, inputs, expected results, and pass/fail criteria for each feature.
    - **Relevance**: Test cases will be created for key functionalities such as verifying employee banking details, validating employee education records, ensuring secure login/logout, and confirming the accuracy of address management. Each case will help check that the features work correctly and meet the specified requirements.
- **Test Scripts (for Automation)**
    - **Description**: Automated test scripts that execute predefined test cases using testing tools like Selenium or other automation frameworks.
    - **Relevance**: Automated test scripts will be developed to speed up testing for repetitive tasks like login/logout testing, performance checks under load, and validating user interfaces across browsers and devices. This will reduce manual testing efforts and ensure consistent execution of tests.
- **Test Data**
    - **Description**: Data used for testing, including employee details, bank account information, education records, and address information.
    - **Relevance**: Proper test data will be created for each feature, such as employee profiles, banking details, addresses, and educational history, to simulate real-world scenarios. This will ensure the application handles real data correctly and securely without any data loss or corruption.
- **Test Summary Reports**
    - **Description**: A high-level summary of all the testing activities, including the overall testing process, major test results, and any issues or concerns identified during testing.
    - **Relevance**: The test summary report will summarize the testing progress for the **Amamder HR Web App**, covering features like bank management, education tracking, and login processes. It will highlight whether the app met the functional and non-functional requirements and will be shared with stakeholders for review.
- **Defect Reports**
    - **Description**: Detailed reports that describe any issues or bugs found during testing, including severity, steps to reproduce, and suggestions for fixes.
    - **Relevance**: Defect reports will be generated if any issues arise in the app during testing. For instance, issues with incorrect bank data handling, failure of the login process, or issues with updating employee addresses. These reports help developers prioritize and resolve critical issues before the app is released.
- **Final Test Report**
    - **Description**: A complete report that includes a summary of all tests conducted, results of each test, defect status, and final conclusions about the quality of the **Amamder HR Web App**.

- **Relevance**: The final test report will provide a comprehensive view of the testing results for the **Amamder HR Web App**. It will confirm whether all key features, such as employee banking, education, and address management, work as expected and whether the app is ready for production. This report will be the basis for the decision to release the app.

8. Risks and Mitigation

Risk: Delays in Development Could Impact the Testing Schedule

- **Description**: If development of key features such as employee banking, education records, or login/logout functionalities is delayed, it can push back the testing schedule and hinder the timely completion of the testing phases.
- **Mitigation**:
  - Conduct regular progress meetings with the development team to track feature completion and resolve bottlenecks early.
  - Adjust testing timelines accordingly and prioritize testing critical features first, especially those that are necessary for business operations.
  - Consider adopting an agile testing approach, where testing is done in smaller increments aligned with the development sprints to minimize delays.

Risk: Unavailability of Key Resources (e.g., Testers, Test Environments)

- **Description**: Unforeseen circumstances, such as team member availability or lack of necessary test environments, can lead to resource constraints, impacting the ability to complete the testing phases on time.
- **Mitigation**:
  - Cross-train team members to ensure they have the necessary skills to step in when required, ensuring testing activities can continue even if a key resource is unavailable.
  - Secure backup resources, either by outsourcing some testing tasks or bringing in additional team members to assist with time-sensitive testing phases.
  - Set up and maintain multiple test environments (e.g., staging, development, production) to ensure that testing can continue even if one environment faces issues.

Risk: Incomplete or Incorrect Test Data

- **Description**: If the test data (e.g., employee records, banking details, address info) is incomplete, incorrect, or unrealistic, it may result in inaccurate test results or missed issues during testing.

- **Mitigation**:
    - Create comprehensive and realistic test data that closely mirrors actual employee information, banking details, and educational records.
    - Perform data validation checks before testing begins to ensure completeness and accuracy of test data.
    - Set up automated scripts to generate consistent and valid test data if needed.

Risk: Compatibility Issues Across Browsers and Devices

- **Description**: The **Amamder HR Web App** might not function or display correctly across all browsers and devices, leading to a poor user experience and potential customer dissatisfaction.
- **Mitigation**:
    - Ensure thorough cross-browser and cross-device testing using tools like BrowserStack to test the app on multiple browsers (Chrome, Firefox, Safari, Edge) and devices (desktops, tablets, smartphones).
    - Prioritize the most commonly used browsers and devices for employees, HR staff, and administrators to ensure they are fully supported.
    - Address any compatibility issues early by involving the development team in fixing the issues as they arise.

Risk: Security Vulnerabilities

- **Description**: The **Amamder HR Web App** may have security flaws that could expose sensitive employee data, such as bank account details, addresses, and educational records, making the app vulnerable to cyberattacks.
- **Mitigation**:
    - Conduct regular security assessments, including penetration testing, to identify and fix potential vulnerabilities.
    - Ensure strong encryption methods are used for sensitive data and implement secure authentication protocols for login/logout.
    - Stay updated with security best practices and patch any known vulnerabilities as soon as they are identified.

Risk: Lack of Effective Communication Between Teams

- **Description**: Miscommunication between the development, testing, and HR teams can lead to misunderstandings of the requirements, missed deadlines, or incomplete testing coverage.
- **Mitigation**:
    - Hold regular communication sessions between all stakeholders to clarify requirements, share progress updates, and address issues as they arise.

- Use project management tools (like Jira or Trello) to track tasks, monitor progress, and ensure everyone is aligned on project goals.
- Create detailed documentation for all processes to ensure clear expectations for both the development and testing teams.

Risk: Insufficient User Feedback During Testing

- **Description**: Not gathering enough feedback from actual users (HR staff, employees) during testing could result in missing important usability issues.
- **Mitigation**:
  - Conduct user acceptance testing (UAT) with actual users to ensure the app meets their needs and provides a smooth user experience.
  - Include feedback loops during the testing process to gather user insights and make necessary adjustments.
  - Prioritize usability testing for critical features like the employee self-service portal and HR dashboards to improve user satisfaction.

9. Entry and Exit Criteria

Entry Criteria

The following conditions must be met before testing begins:

1. **Development Complete and Code Freeze Announced**
   - All core development tasks and feature implementations (such as bank management, education management, and login/logout functionalities) must be completed. There should be no further changes to the codebase, and any new features should be in their final form to allow stable testing.
2. **Test Environments Set Up and Ready**
   - All necessary test environments (development, staging, and production) must be fully configured and ready for testing. This includes the availability of the required server infrastructure, databases, and external integrations needed for testing.
3. **Test Data Prepared**
   - Relevant test data (e.g., employee details, banking information, education records, and address data) must be created, validated, and ready to be used for testing the app. This includes ensuring that the data is accurate, complete, and realistic.
4. **Test Cases Reviewed and Approved**
   - Test cases, covering all critical features and functionalities, must be reviewed and approved by the project stakeholders and QA team. These test cases should clearly define the testing objectives and expected results for each feature of the app.

Exit Criteria

The following conditions must be met before testing is considered complete:

1. **All Critical and High-Priority Test Cases Executed**
    - All critical and high-priority test cases, which cover core functionality such as the banking features, employee education records, address management, and login/logout process, must be executed and passed successfully. This ensures the app's primary functions are working as expected.
2. **No Open Critical or High-Severity Defects**
    - There should be no unresolved critical or high-severity defects in the system. Any issues that impact key functionalities, such as employee data management, banking transactions, or security vulnerabilities, must be fixed before testing can be completed.
3. **Test Coverage Meets the Target**
    - The test coverage (i.e., the percentage of code, features, and workflows tested) must meet the predefined target, which is typically set at 90% or higher. This ensures comprehensive testing of the app's features and prevents any major gaps in coverage.
4. **Test Summary Report Reviewed and Approved by Stakeholders**
    - The test summary report, detailing the overall testing process, results, and any outstanding issues, must be reviewed and approved by the relevant stakeholders (e.g., project managers, development team, and product owners). This report should summarize all test activities and validate whether the app is ready for release.

10. Estimation Techniques

The effort required for testing the **Amamder HR Web App** is estimated using a combination of **historical data** from similar past projects and **expert judgment** from experienced team members. The goal is to provide a realistic and efficient estimate for the required testing effort.

1. **Historical Data**
    - **Description**: Past projects, particularly e-commerce and HR management systems, serve as benchmarks. By analyzing the testing efforts, timelines, and complexities of similar applications, historical data helps provide an initial estimate for the **Amamder HR Web App**.
    - **Relevance**: Since this app involves several key components, such as bank management, education management, and secure login/logout systems, the testing effort for similar features in previous e-commerce or HR apps is reviewed to form the basis for the current estimate.

2. **Expert Judgment**
    - **Description**: The project estimations are also based on the knowledge and expertise of senior QA team members, who are familiar with the specific challenges and scope of the **Amamder HR Web App**. The team considers the complexity of the features (e.g., employee data management, payment integration) and the level of testing required (e.g., functional, security, and performance testing).
    - **Relevance**: Expert judgment allows the team to account for specific nuances in the project that may not be captured by historical data alone, such as integration with external systems or the specific security measures needed.

Estimated Effort

Based on the above techniques, the total estimated effort for testing the **Amamder HR Web App** is approximately **300 person-hours**. This estimate is broken down as follows:

- **Test Planning**: 30 hours
- **Test Case Design**: 60 hours
- **Test Execution**: 120 hours
- **Bug Fixing and Retesting**: 50 hours
- **Regression Testing**: 40 hours

11. Test Case Prioritization

High Priority

1. **User Registration and Login**
    - **Description**: Ensuring that users (employees, HR staff, and admins) can securely register, log in, and log out of the system.
    - **Reason for High Priority**: This is critical for access control and overall security. Any issues here could block access to the entire application.
2. **Employee Data Management (Bank Management, Education, Address Management)**
    - **Description**: Verifying that employee data, such as bank details, educational history, and address, are accurately recorded, updated, and retrieved.
    - **Reason for High Priority**: These are core functionalities of the **Amamder HR Web App** and must be thoroughly tested to ensure employee data is handled correctly and securely.
3. **Checkout Process (for Subscription or Services)**
    - **Description**: If the app has subscription-based features or any paid services, the checkout process must be smooth and error-free.

- **Reason for High Priority**: A functional checkout process is essential to ensure users can subscribe to the platform or make payments successfully.

4. **Payment Gateway Integration**
   - **Description**: Testing the integration of payment systems to ensure that payments are processed correctly and securely (e.g., handling of employee subscriptions or service payments).
   - **Reason for High Priority**: Payment issues could lead to lost revenue or data breaches, making this a critical area for testing.
5. **Security Testing**
   - **Description**: Testing for vulnerabilities such as SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF).
   - **Reason for High Priority**: As the app deals with sensitive employee data, ensuring that it is protected against potential attacks is crucial.

Medium Priority

1. **Product Search and Filtering**
   - **Description**: If applicable, ensuring that search and filter functions (for employee records, HR reports, etc.) return accurate and relevant results.
   - **Reason for Medium Priority**: Although important for user experience, this functionality is secondary to the core features like login and data management.
2. **Shopping Cart Functionality (for Paid Services or Subscriptions)**
   - **Description**: Testing the shopping cart to ensure that items (e.g., subscription packages, additional services) are correctly added, edited, and removed.
   - **Reason for Medium Priority**: Important for users purchasing services, but non-essential compared to core business features like data management.
3. **Order History and Tracking**
   - **Description**: If the app has paid features or subscriptions, users should be able to view their order history and track progress.
   - **Reason for Medium Priority**: A useful feature, but non-critical for the initial release or operation of the platform.

Low Priority

1. **User Account Management**
   - **Description**: Features related to users managing their profiles, such as updating their personal details or changing passwords.
   - **Reason for Low Priority**: While important, this is less critical than core features like login and employee data management. This can be tested after higher-priority features are complete.
2. **UI/UX Testing (Aesthetic Issues)**

- **Description**: Testing for consistency in design, layout, and navigation, focusing on non-functional aspects of the user interface (UI).
- **Reason for Low Priority**: Aesthetic issues can be fixed post-launch or in subsequent versions. These do not impact core functionality but are important for user satisfaction in the long term.