



Manual Testing

Section 1 -

- **Project Discussion- SchoolFee, BloodBank, (SRS, User wise, Module, Screen/ Scenarios/ Feature), Screen Development**
- **Test case for Login, SignUp, Search, Bus Booking**
- **White Box Testing – Unit Testing and Integration Testing, Black Box – Integration, System Testing**
- **Bug Reporting, LifeCycle and Test Teport**
- **API Testing**
- **Performance Testing**

- *1. Discuss and analyze the requirements of two projects: SchoolFee and BloodBank. Create a Software Requirement Specification (SRS) document for each project, including user-wise modules and screens/scenarios/features.
 - *2. Develop at least three screens for each project based on the requirements identified in the previous task. These screens should be implemented using a suitable technology or prototyping tool.
 - *3. Develop test cases for the login, sign-up, search, and bus booking functionality of both the SchoolFee and BloodBank projects. Include positive and negative test cases, covering different scenarios and user interactions.
 - *4. Apply test design techniques such as Boundary Value Analysis and Equivalence Class Partitioning to identify and design test cases for specific scenarios. Perform field validation testing and ensure code coverage by executing test cases.
-
- #1. Create general test scenarios that cover common functionalities across both projects. Develop test scenarios based on user interactions, system behavior, and expected outcomes. Ensure that all critical paths and edge cases are covered.
 - #2. Perform unit testing and integration testing using white box testing techniques for the SchoolFee and BloodBank projects. Execute black box testing at the integration and system levels. Identify and report any bugs or issues encountered during testing.
 - #3. Learn about the bug reporting lifecycle and use a bug tracking tool like Jira to report and manage bugs found during testing. Understand the process of test reporting, including test execution status, defects identified, and overall test coverage.
 - #4. Use tools like Postman to perform API testing for the SchoolFee and BloodBank projects. Test the functionality, performance. And security of the API endpoints. Learn about JSON structure and create test cases for API operations using Postman.
 - #5. Learn the concepts of performance testing and use JMeter to test the performance of the SchoolFee and BloodBank projects. Create test scenarios execute load and stress tests, and analyze the performance metrics and results.