Ex.No: 1
5.6.2026

1: Basic Test Plan Document for a Sample Project

Aim:

To prepare a test plan document for an online book store application by defining its scope, objectives, resources & schedule

Procedure:

1) Select the project to be tested
2) Define the scope of testing
3) Identify testing objectives
4) Allocate testing resources
5) Prepare the testing execution schedules
6) Document the test plan

Test Plan:

Project Name:

Online book store Application

Scope:

* User Registration
* User Login
* book Search
* Add to Cart
* Order Placement

Objectives:

- Verify application functionality
- identify and Report defects
- Ensure Requirements are met
- improve Software quality

Resource	Quality
Test Manager	1
QA Engineer	2
Developer	2
Test Environment	1

Schedule :

Activity	Duration
Requirement Analysis	2 Days
Test Case Design	13 Days
Test Execution	5 Days
Defect Reporting	2 Days
Retesting	2 Days.

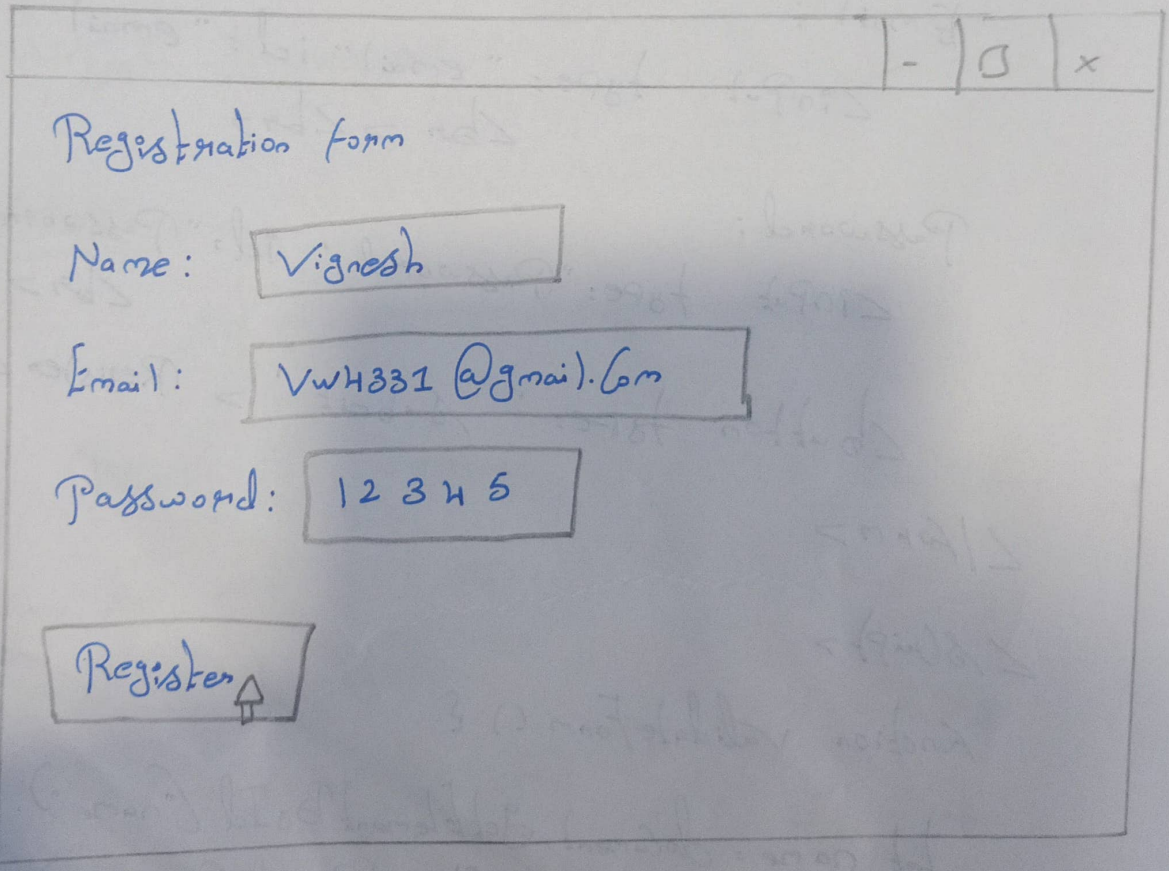
(Registration form for online book Store Application)

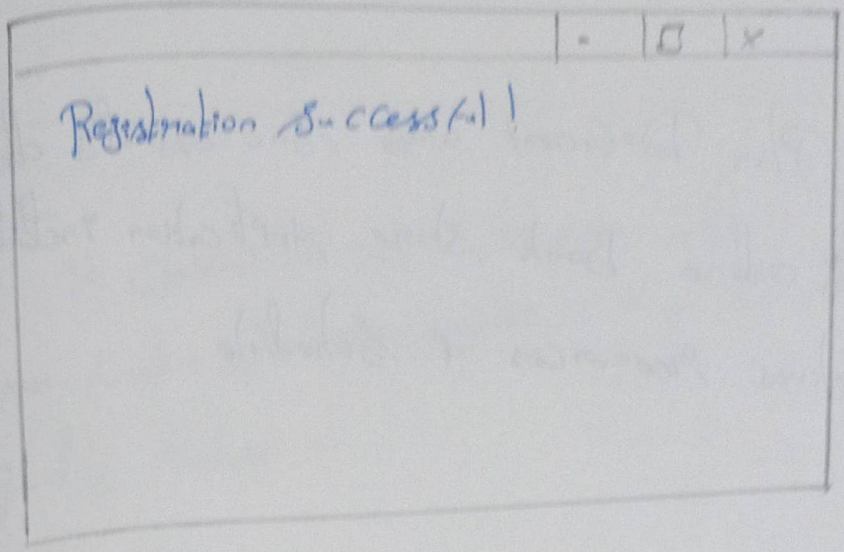
Code:

```
<html>
<head>
<title> Registration form </title>
</head>
</html>
<body>
<h2> Registration form </h2>
<form onSubmit = "return validateForm()">
Name:
<input type="text" id="name"> <br>
<br>

Email:
<input type="email" id="email">
<br><br>

Password:
<input type="password" id="Password">
<br><br>
<button type="submit"> Register </button>

</form>

<script>
function validateForm() {
let name = document.getElementById("name").value;
let email = document.getElementById("email").value;
let password = document.getElementById("Password").
value
```

```
if (name == "" || email == "" || password == "") {
    alert ("fill all fields");
    return false;
}
alert ("Registration Successful!");
return true;
}
</script>
</body>
</html>
```

Output:





Test Case Table:

SNo	Input Data	Type of test	Expected Output	Actual Output	Result
1	Name: Vignesh. Email: v@gmail.com password: 12345	Valid	Registration Successful	Registration Successful	Pass
2	Empty fields	Invalid	Error: fill all fields	Error: fill all fields	Pass
3	Name only	invalid	Error message	Error message	Pass
4	Invalid Email (abc@)	Invalid	invalid email	invalid email	Pass
5	Password empty	invalid	Error message	Error message	Pass
6	Special Character in name (@##)	invalid	Error	Error	Pass
7	Short Password (123)	invalid	Password too short	Password too short	Pass
8	Duplicate Registration Data	Invalid	Already exists	Already exists	Pass
9	Valid Data again	Valid	Registration Success	Registration Successful	Pass

Result :

The test Plan Document was successfully designed for the online Book Store Application including scope, objectives, resources & schedule

Ex:2 Classify Software Requirements & Map them to Quality Factors

Aim:
To Classify Software Requirements in to functional and Non-functional Requirements & Map them to relevant Software quality factors.

Procedure:
1) Study the Software Requirements Specification (SRS)
2. Identify functional Requirements.
3. ~~Classify~~ identify Non-functional Requirements
4. Classify the Requirements based on there frame.
5. Map each Requirements to the appropriate quality factors.
6. Analyze the relationship between requirements & quality factors

Requirement classification & quality Mapping:

functional Requirements:
functional Requirements
functional Requirements describe what the System should do

Req ID	Functional Requirement	Quality factor
FR1	User should be able to register an account	functionality
FR2	User should be able to login using credentials	functionality
FR3	User should search for products	functionality
FR4	User should add products to cart	functionality
FR5	User should place orders online	functionality

Non-functionality Requirement:

It ~~shoul~~ describe how the System should perform

Req ID	Non-functional Requirement	Quality factor
1	System response time should be less then 3 seconds	Performance Efficiency
2	System should be available 99.9% of the time	Reliability
3	Application ~~Passwords~~ should support multiple browsers	Compatibility
4	Passwords should be encrypted	Security
5	User interface should be easy to use	Usability

6 System should be easy to Modify & Maintain — Maintainability

7 Application should work on Windows, Linux & Mac — Portability

Quality Factor Description

Quality Factor	Description
1) Functionality	Ability of the Software to perform Required functions Correctly.
2. Reliability	Ability to the System to operate without failure
3. usability	Ease of learning & using the Software
4. Performance Efficiency	Efficient use of System resource & Responsetime
5) Security	Protection against unauthorized access & Data breaches.
6) Maintainability	Ease of Modification & enhancement.
7. Compatibility	Ability to work with different Systems & environments.
8. Portability	Ability to run on various Platforms.

Result:-

Software Requirements were successfully classified into Functional & NON-Functional Requirements and mapped to their Corresponding Software quality factors. This classification helps ensure that both business needs & quality expectations are addressed during Software development & testing.

Ex:3 Draft an SQA Policy Placement

Aim:

To develop a Software Quality Assurance (SQA) Policy document that defines quality objectives, standards, responsibilities and procedures for ensuring Software Quality.

Procedure:

1. Identify the Quality objectives of the Organi-zation
2. Define roles and Responsibilities for quality Assurance
3. Establish quality Standards & Procedures.
4. Specify Review, testing & defect Management Process
5. Document the SQA Policy.
6. Review and Approve the Policy document.

SQA Policy Document:

1. Purpose:

The Purpose of this Software Quality Assurance Policy is to ensure that all Software Products are developed, tested & Maintained according to established quality standards and Customer.

2. Scope:

This Policy applies to all Software development, testing, maintenance & support activities within the Organization.

3. Quality objectives:

* Delivery high-quality Software Products.
* Minimize defects in Software releases.
* Ensure Customer Satisfaction.
* Comply with Organizational and industry standards.
* promote Continuous Process improvement.

4. Roles & Responsibilities.

Role	Responsibility.
* Project Manager	Ensure Compliance with SQA Policy
* QA Manager	Monitor quality activities & audits.
* Developers	follow Coding Standards & fix defects
* Test Engineers	Design & execute test Cases
* Customers	provide feedback and acceptance Criteria.

5. Quality Standards :

* Requirements shall be clearly documented.
* Coding Standards shall be followed.
* Peer reviews shall be conducted
* Testing shall be performed before release
* Defects shall be tracked & resolved System ati Calls

6. Testing Process :

* Unit testing.
* Integration testing
* System testing
* Regression testion.
* UAT

7. Defect Management :

* All defects shall be logged in a defect tracking System.
* Defects shall be prioritized based on severity.
* Fixed defects shall be retested before Closure
* Defects reports shall be maintained for future - Analysis.

8. Review & Audit:

* Regular quality reviews shall be Conducted
* Internal audits shall verify Compliance with Standards.
* Corrective actions shall be taken for identified issues.

9. Continuous improvement:

* Collect quality metrics regularly.
* Analyze defects and root causes
* implement Process improvements based on findings
* Conduct training programs for team members.

Result:

A SQA Policy Document was Successfully drafted, defining Quality objectives, Standards, responsibilities, testing procedures & Continuous improvements practices to ensure Software Quality.

Exp 4

Implementing the Contract Review Process with a Case Study.

Aim:
To implement the Contract Review Process for an online book store Management System and ensure the Customer Requirements are clearly understood, document and agreed upon before Software development begins

Procedure:
1. Receive the Customer Requirements document
2. Analyze the Project Scope & objectives
3. Review functional & non-functional requirements.
4. identify risks, Constraints & assumptions.
5. verify technical feasibility and resource availability
6. Discuss ambiguities with the Customer.
7. obtain approval from both Customer & development team
8. finalize the Contract review Report

Case Study: Online bookstore Management System.

Project Description:
the Client Requires an Online book store Management System. that allows users to browse books, Search products, Manage shopping carts.

Place Orders & make Online Payments

Contract Review checklist:

Review item	Status	Remarks.
* Project Scope Define	Approved	Clear Requirements Provided
* Functional Requirements Reviewed	Approved	All modules iden-tified
* Non-Functional Require-ments Reviewed	Approved	Performed & Security specifi
* Resource Availability Checked	Approved	Team assigned
* Budget Verification	Approved	with in Client budget
* Schedule Verification	Approved	4 month timeline accepted.
* Technical feasibility Verified	Approved	Technology stack finalized.

functional Requirements

Req ID	Description
FR 1	User Registeration
2	User Login
3	Book Search
4	Add books to Cart
5	Place orders
6	online Payment
7	Order tracking

Non-functional Requirements

Req ID	Description
1	Response time less than 3 seconds
2	99.9 % System availability
3	Secure user authentication
4	Cross-browser Compatibility

Risk Analysis

Risk	impact	mitigation
Requirement changes	High	Requirement freeze after approval
Schedule Delay	Medium	weekly Project monitoring
Resource unavailability	Medium	Backup Resource allocation
Security Issues	High	Security testing & audits.

Contract Review Report:

item	Details
* Customer name	ABC book Distributors
* Project Name	Online bookstore Management System
* Review Date	02-06-2026
* Project Duration	4 months
* Estimated Cost	₹5,00,000
* Review Status	Approved

Conclusion:

After reviewing the Project Requirements Resources, Schedule, budget, & risks the Contract was

found to be feasible & acceptable. Both the Customer & development team agreed on the project Scope and deliverables

Conclusion:

After Reviewing the Project Requirements, resource, schedule, budget and risks the Contract was found to be feasible & acceptable both the Customer and development team agree

Result:

the Contract Review Process for the online book Store Management System was successfully implemented All Requirements, Resources, Schedules, Risks & Project Contracts were Reviewed and approved before Project initiation.

Ex: no: 5

Prepare Software Development and Quality Plans for Project System.

Aim:

To Prepare Software Development Plan and Software Quality Assurance Plan for an Employee Leave Management System.

Procedure:

1) Identify the project requirements.
2. Define the Software development activities.
3. Allocate resources & responsibilities
4. Establish project milestones & schedules.
5. Prepare the Software Quality Assurance Plan.
6. Define testing, Review & Quality Control activities.
7. Document the plans & obtain Approval.

Project: Employee Leave Management System.

Description:

the Employee Leave Management System is designed to automate leave Requests, approvals leave

balance tracking and Report Generation for employees & Managers

Software Development Plan:

1. Project Objectives:
   - Automate employee leave Management
   - Reduce Manual Paperwork
   - improve leave tracking & Reporting
   - Ensure Secure access to leave records

2. Project Scope:
   The System includes:
   - Employee Registration/login
   - Leave Application
   - Leave Approval/Rejection
   - Leave Balance Management
   - Leave History tracking
   - Report Generation.

3. Project Team:

Role	Responsibility
* Project Manager	Project Planning & monitoring
* System Analyst	Requirements Analysis.
* Developers	Coding & implementation.
* Test Engineers	testing & Quality Assu

* Database Administrator Database management.

4. Development Schedule

Activity	Duration
Requirement gathering	1 week
System Design	1 week
Coding	3 week
Testing	2 week
Deployment	1 week
Maintenance	Ongoing

5. Development methodology
   * Agile Development Model
   * Iterative Development
   * Continuous feedback

Software Quality Assurance Plan

1. Quality objectives:
   * Deliver defect-free Software
   * Meet all Customer requirements
   * Ensure System reliability & security
   * improve user satisfaction.

2. Quality Standards:

* Follow Coding Standards
* Maintain Complete documentation.
* Conduct Peer Coding Reviews
* Execute test cases before release.

3. Review Activities:

Review type	Conducted by
* Requirement Review	Project Managers & Analyst
* Design Review	Development Team
* Code Review	Senior Developers.
* Test Case Review	QA Team.

4. Testing Plan:

Testing Type	Purpose.
* Unit testing	Verify individual modules.
* Integration testing	Verify module interactions
* System testing	Validate Complete System
* Regression testing	Verify existing
* User Acceptance testing	Validate Customer Require-ments.

5. Defect Management Process:

New
↓
Assigned
↓
Open
↓
Fixed
↓
Retest
↓
Verified → Closed

6. Quality Metrics

Metric	Target
* Defect Density	< 2 Defects / kloc
* Test Case Pass rate	> 95%
* Requirement Coverage	100%
* Critical Defects	0 before Release.

7. Risk Management

Risk	Mitigation
1) Requirement Changes	Requirement Review Meeting
2. Schedule Delays	Regular Project Tracking
3. High Defect Rate	Early Testing & Reviews
4) Security Issues	Security Testing

Result:

Software Development Plan and Software Quality Assurance Plan were successfully prepared for the Employee leave management System, defining project scope development activities, Quality objectives, testing strategies & Quality Control measures.

Ex: No: 6

Prepare V&V Plan with defect removal for library Management System.

Aim:

To Prepare a Verification & validation (V&V) Plan and establish a defect Removal Process for the library Management System to ensure Software Quality & reliability

Procedure:

1. Gather & analyze the System requirements
2. Define verification activities for each development Phase
3. Define validation activities and ~~Removal~~ Process for testing the System
4. Establish a defect identification & Removal Process
5. Prepare the V & V Plan document
6. Review and finalize the Plan.

- Requirements for Library Management System:

Functional Requirements

Req ID	Description
1	User should be able to login using valid Credentials.

2. Librarian Should be able to add new books
3. Librarian Should be able to update book details
4. Librarian Should be able to delete book records.
5. Members should be able to search books.
6. Librarian Should be able to issue books to members.
7. System & Librarian should be able to return books from members.
8. System should calculate overdue fines automatically
9. System should maintain member record
10. System should generate library reports.

Non-Functional Requirements ID	Description.
1	System response time should be less than 3 seconds
2	System should be available 24x7
3	User data should be secure and protected
4	System should support multiple browsers.
5	System should be easy to maintain & update.

Verification Plan

Phase	Verification Activity
Requirements	Requirements Review
Design	Design Review
Development	Code Review
Test Case	Test Case Review
Documentation	Documentation Review

Validation Plan

Testing Plan	Purpose
* Unit Testing	Validate individual modules.
* Integration Testing	Validate module interactions
* System Testing	Validate Complete System
* Regression testing	Validate Existing Functionality
* User Acceptance testing	Validate Customer requirements.

Defect Removal Process :

Step	Activity
1	Defect identification
2	Defect logging
3	Defect ~~logging~~ Analysis
4	Defect Anal Assignment
5	Defect fixing
6.	Retesting
7	verification
8	Defect closure

Defect Severity Classification :

Severity	Description
Critical	System Crash or data loss
High	Major functionality failure
Medium	functional issue with works around
Low	Cosmetic or UI issue

Quality metrics :

Metric	Target
1. Requirement Coverage	100%
2. Test Cass Pass Rate	>95%
3. Defect Removal Efficiency Critical Before Release	>95%

Result :

The Requirements for the library Management Syste were identified and a verification & Validation Plan with defect removal procedures was successfully prepared to ensure Software quality and compliance with user Requirements.