```
Testing Terminologies----->
2 types-----> service based or product based

Manual Test engineer---->varification--->
        Manually collects the tc data
        he can write tc in excel or testrail or sharepoint
        TC is written by Human
        Report will generated manual

Automation Engineer----->validation--->***
    unittest framework**----> assert actual == expected(Low Level
Design,High Level Design)
        in script(python,java,c++,c#)---> selenium webdriver+python
                                    ----> mysql,orale,mysqlllite3+python
                                    ----> PostMan , requests+selenium or
robot
        Report will generated automatically---> HTML or XML
        Selenium,

Developer-----> implementation of Function
TL----> team lead---> handle the team---> senior person 6+
Manager---->Technical or non technical---->4-7 teams
Engineering Director---->

**SM(Scrum master) ---->daily task asssigned /TL/Manager/Senior Person
                     ----> ejira tools

Defect---->issue, indence---> identify by test engineer
Bug ---->when developer accept the defect
error --->any problem in code
        syntax error--->
        Runtime error----> exception hnadling
Failure --->End user getting the defect
tcs(service)---->sbi yono(product)---->customer, sbi customer
            ---> maharashtra
            ---->HSBC

1 Project----->Requirements will come from customer(client)
        ----> customer(technical or nontechnical) and his people
2 Product----> for 1 project ----> n product
        ---->Requirements will come from various customer

3 Application-----> Group of program designed for the customer
             -----> web based application (django,flask)

4 AUT----> Application Under Test
    after the design and coding there is Testing
    before the deployment

5 Quality---> justifications of Requirements
        ----> absence of defect
6 Defect--->devionation from the Requirements
7 SRS---> System Requirements specifications---->
```

lld and hld----> actual requirements Documents----> sharepoint or confluence page
8 BDD---> Business Design Documents
----> before SRS this documentation is happen

9 Mock Ups---->in design phase
screen shots provided during design

10 Use case---->description of flow of your product

11 Test case---->how test the application and how to check it
12 Test Data---->data required for TC
13 Version COntrol---- GIT and GIHUB
14 Check Out----> taking the documentation for edit (do the changes)
15 Check In--->relase the documentation after the edit
16 Release Notes----> developer----> doct string
---> for every function
17 Build ----build is a version software
18 Release----> will not test, ready to Deliver to the customer
19 Delivery ----handover our product/project/application to customer
20 Test Deliverable---->during testing the documentation is prepared that we handover to customer
21 Test summary report---->
total tc
total pass
total failed
22 Traceability matrix---->
bi-directional matrix---> TC and its requirements
23 Productivity---> hr produced by team member----> product
24 Variances ---->difficence between what was planned and what exactly we are testing
25 Estimations ---->time for specific work, cost estimation---> cost for specific work
26 Escalation ----> taking issue in next step
27 Roles and Resposibility--->
28 Entry Criteria----> is used to determin when to test activity
29 Exit Criteria---->
30 Status Call ------> every day there is status--->
yester day what we done??
today what we can do??
31 MOM -----> Minutes of Meeting-----> idea or notes from the meeting
32 One to One--->it is a face to face meeting
33 Appraisal --->Dec / Jun
34 Rating--->given for each team member
35 Time sheet --->for day we need to fill / Weekly
Test Case Prepare--->
Weekly meeting--->
Requirements understanding --->
Debugging --->

36 resource billable ----> team member---> ctc
37 Non Billable ---> Bench----> no task---> attached to any project
38 OOO ----> out of office---> leave
39 planned leaves----> manager for every month

40 un planned leaves----> emergency leave
41 PTO ---> Paid time off
    16 Planned leaves (4 per quaters) and 5-10 sick leave
42 Notice Period --->at time switch
                1month/2month/3month
43 Bench marks --->finalising the document or any process
44 CTC---->cost to the company----> Package(fixed salary+variable salary)
45 Variable pay--->based on your performance this salary is there
46 Basic Salary --->
47 Gross Salary --->
48 Demo----> product information/product presentation to client
49 POC ---> point of contact----> service based to product
        ----> proof of concep----> agreement of proof
50 Pipeline---> in progress/in process
51 project Sign In----> Project is finalised / team will work on the
project
52 SDLC---->**
    Requirements gatherings
    Analysis
    design
    Coding/Development
    Testing
    Development
53 STLC--->**
54 Service level agreement----> 3rd party organization/contract
55 Code of conduct agreement---->at the time of onboarding
---->
56 Tools----> manual testing--->testrail,excel,sharepoint
57 Technologies ---> python
58 Bug life cycle**---->for bug tracking

Software Testing---->
    Verification(white box,unittest,static testing) and
validation(blackbox,pytest,dyanamic testing) of any application

Defect----> issue,incident ----> Test engineer(Manual+automation)
        ---> deviation between actual and expected result(LLD,HLD)
Bug ---> if defect is accepted by developer(django,flask)
    jira ---> bug tracker

error---> problem in Program/application   --->
    ---> not getting proper result
    exption -->

Failure---> end user is getting some defect

Activity--->
Coding---> Done by developer(whitebox)
**Testing---> Done by Tester(manual or automation)(whitebox or blackbox)
Defect reporting----> Done by Tester(manual or automation),jira(logs,SS)
Debuging --->Done by developer (pdb), time,logging module(info,error)
Bug fixing---> Done by developer

Testing team----> 3 automation+1manual engineer
manual Tester--->Excel or TestRail Tool
automation Tester---> Python ---->
Selenium(UI,DB,API),fastAPi,Robot(API), Userframework (Medley)

Role AND Responsibility of test engineer---->

1. Review on BRS(Business Requirements specification),FRS(Function
Requirements specification)
2. Identify test senions of allocated module (owner)
TS----> what to test in test
TC----> How to test, how Verify and valide
3. writting the TC
facebook login page---->
TS1-->login validation
    TC1--->verify the valid login data(username and pass(8))
    TC2--->verify the unvalid login data
    TC3--->verify the login with data
TS2--->signup validation
TS3--->reset validation

4. Implemente the automation tc
5. Review all automated TC and automation framework
    ----> TO check the correctness/completeness of application
6. Execution of TC and Test scripts(modules)
    Jenkins----> CICD
    pipeline----> tc1,tc2,tc9,tc5

7. Defect reporting and identification of defect
    jira-----> description----> logs or ss
            ---> usename empty
            ----> password empty
            ----> click login   ----> we are not proceed to timeline

8. Performing the re-testing
9. work with developers

```
pip -----> python install packaage
pip install package_name
pip install lab

pip install selenium

testing---> varification and validation

Manual Testing---->
                all data set is manually created
                TC is written by Human / Manual Test engineer--->
Excel/sharepoint/Test Rail
                Manual report is generated
                Excel/sharepoint/Test Rail
TC report----> total TC
            ----> Failed TC
            ----> Passed TC  -----> assert expected and actual output
            ----> % of passing TC  -----> 90% + passed


Automation Testing-->
            tools or script(python,java,.net) we are write our TC
            Report is generated automatically(HTML,XML)
            selenium webdriver
            pytest,unittests
            pymysql
            requests

    TS----> what to Test
    TC---> How to test

Unit Testing---->functional testing
            ----> unittests
            ----> white box

Itegration Testing ----> all functionality of application
                    ---->final testing
                    ---> end to end
                    ----> QA/Test engineer respo

system testing---> test overall application

load testing---->max end user testing

black box testing---->no need of code

white box testing---->code knw----->

sanity testing----->during the release check the main functional
CRUD
smoke testing---->major test cases are run in smoke testing
                ----> test case are running or not
                ----> product is testable  or not
                ----> build(code)
```

209---> 100
109---->50
50----> % final

regration testing----> all test case run repeatdly

mockey testing ----> Random testing done on any application/product
                ----> data is not present
                ----> predefine rules not present
                ----> 10000
                ----> random IP address
                -----> name,ip,DNS
Automative domain -------> 4 type cerficates
cleint---->

Performance testing-----> 3.00 sec

hp,cisco,dell,samsung---->4 devices

white box----> we have code as well UI part
        ----> developer---> unittest

black box----> code is not available

Test senarios----->what to test
Test case---->how to test
Test suite----->group of test cases----> Itegration---> regration testing
running the tc / trigger the job

report generation---->
    line

bug--->

New---->
Duplicate--->
fixed---->
Re-Open---->

Bug life cycle---->
6 steps---->
1 test engineer find the Bug----> New or open (status)(BUGID)
2 Bug will assign to the Dev. Manager
            Dev.Manager---> if not valid----> rejection
3 Test again check ----> similar
        if yes----> Duplicate bug(BUGID)
   Defect
4 Once bug is fixed---->status ---> fixed(BUGID)
5 Re-test case---->if tc is passed----> closed
6 Re-test case---> if tc is failed---->Reopen and assign to the developer

SDLC and STLC---->
Test Rail---->

FAN---->

1 Verify the fan type----> table fan or celling fan ---black box testing Pass/Fail
2 ON /OFF ----> button is working or not
3 Speed reguation using the fan regulator   ----> load testing
4 How much volatage is using FAN
5 Verify the Heating Fan   --->Performance testing
6 verify the number of blades (3)
7 Verify the material type for FAN or Blades
8 Verify time----> to attain maximum speed   ----> sanity
9 verify time ---> to attain minimum speed
10 verify fun will rotate in left to right or right to left
11 verify the color of fan
12 FAN 5hr ----> heating

Water Bottle---->
1
2
3
4
5
6
7
8
9

Search Functionality

ATM Login Machine

Movie Booking

SDLC---->
STLC---->
Manual Testing Tools--->

Mini-TC documentation**
Python Mini**

SDLC---->software development life cycle
for every application go with this cycle....

1 Requiement collection---->
    B.A(own) <----> Client(other)
        SLA----> service level agreement
        BRS---->Business Requiement specification
        SRS----> software Requiement specification
        URS---->user Requiement specification

        Feasibility study---->
            Time--->?
            **budget and cost ---->ctc--->
            functionality ---->
            client expected---->

2 Requiement analysis---->FRS---> functional Requiement specification
    system analyst---->
    button,components-->

3 Design ----> Design architect
    HLD LLD------> they function_name,class_name,module_file
            -----> test_name,browser,report
    GUI,DB,application

    abstraction------>only declaration no implem
    in child

    from abc import ABC,abstractmethod
    class Calculator(ABC):
        @abstractmethod
        def add():
            pass
        @abstractmethod
        def sub():
            pass

4 Codings ----> Developer
    Function , ***unittest (white box)
    1 testing and then 2 development

    abc.py
    class Modifiled_Calculator(Calculator):
        def add(a,b):
            c=a+b
            return c

```
        def sub(a,b):
            c=a-b
            print(c)
    Test* or *Test
    test_* or *-test

    test_abc.py
    import abc
    from unittest import Testcases
    class Test_Calculator(Testcases):
        def __init__(self):
            obj=Modifiled_Calculator()
        def test_add():
            result=Modifiled_Calculator.add(10,20)
            assert 30==result     #30==30

        def test_sub():
            result=Modifiled_Calculator.sub(10,20)
            assert -10==result

** 5 Testing ----> QA,UAT,Black box
   Failed with any input
validate
add('str'+'str')---> add(num1+num2)=num1+num2
sub()---->
    verification and validation
6 Release / maintains ---->Deployment Team


*******
STLC---->software Testing Life Cycle
1 Requiement analysis ----> Test Manger
2 Test Planning----> Test Team ----> Team Lead
    what to test---> TS
    how to test---> TC
    when to test--->time
    who will test--->employee110@in.com
3 Test case development----->
    manual testing---->
    automation testing--->script (python)
    test writing
4 Environments setup----??
    fixture(unittest,pytest)---->clean up and setup
5 Test Execution---->
    test run----> tc triggered
    report generated---->
    after trigging jobs ---- 4 - 7 hr
        mail all scrum
        tc report----->
        total tc
        fail tc
        pass TC
        %
```

6 Test Cycle closure

SDLC----> Coding part----------> white box
STLC---->

Manual TC-----> Test Rails ----

UI Testing
Database Testing
API testing
Mobile testing

id testcase
title testcase
    description---> step by step

monkey testing---->

Agile methodology----->
sprint---> collection of task() -----> 21days
0-2exp ----> 3-4
2-4exp ---> 5+
4+exp -----> 10+ UI
    2-----> 1
    2---->1

22nd day----> different
3-4 Reaserch work-----> i am going through database

Selenium----->
 2-3 browser----->drivers()
 excutable_path=

 Developer
 technology---->
 Manual----> closed, TC, TestRail
 automation---> Selenium
 core python developer