1) WHAI IS SDCL?

- SDLC stands for Software Development Life Cycle,

- a structured process used by development teams to create, test, and maintain software.

- It provides a map for planning, building, and deploying software, ensuring a systematic and efficient approach to software development.

- SDLC helps to minimize project risks, meet customer expectations, and produce high-quality software.

- 1. Request gathering and analysis

: Identify, collection, and analyze the needs of software testing from clients.

2. Design

: Selection for uses of database and Frame work

: System Architect of project  
 : Future scope

3. Development

: Which developed by developer (ex. coding for respective software)

4. Testing

: Finding a bug in a software or website

5. Deployment

: Making software or product or update available in particular environment  
 : Ensure making software is useful or suitable after development

6. Maintenance

: Process of maintain of software after release and fulfil all the user needs

2) WHAT IS SOFTWARE TESTING?

- Software testing is about making sure a software application works as it should and is free of errors or bugs.,

- It’s a like checking quality of software throughout finds for a bug or any kind of errors before its release.

3) WHAT IS AGILE METHODOLOGY?

- Agile methodology is involves incorporating testing throughout the development lifecycle,

- It's a collaborative, iterative approach where testing isn't a separate phase but an integral part of each iteration

- In which we can test software constantly while developing is in progress.

4) What is SRS

* Software Requirements Specification (SRS) for eCommerce Website  
   1. Introduction  
   1.1 Purpose  
   This SRS defines the requirements for an eCommerce website that allows users to browse, search, and purchase products online. The system will support user registration, secure payment processing, order tracking, and administration features.

1.2 Scope  
 The website will serve as an online platform for sellers to list their products and for customers to make purchases. It will include:  
 User authentication  
 Product catalog  
 Shopping cart  
 Order management  
 Payment integration  
 Admin panel  
 1.3 Definitions  
 User: A customer who browses or makes purchases.  
 Admin: A backend user managing the system.  
 Product: An item listed for sale.  
 Cart: Temporary storage of selected items before purchase.  
 2. Overall Description  
 2.1 Product Perspective  
 The eCommerce website is a standalone web application built using a modern tech stack (e.g., React, Node.js, and MongoDB).  
 2.2 Product Functions  
 Register/login/logout  
 Search and filter products  
 Add/remove products to/from cart  
 Checkout and process payments  
 View order history  
 Admin dashboard for managing products, users, and orders  
 2.3 User Classes // use case // website flow - client   
 uml - unified modelling lang - diagram   
 1 use case diagram - actor   
 2 class diagram - table   
 3 activity diagram   
 4 sequence diagram -  
 5 package diagram   
 6 deployment diagram   
 Guest Users: Can browse and search products.  
 Registered Users: Can purchase, manage orders, and write reviews.  
 Administrators: Can manage all content and users.  
 2.4 Constraints  
 Web-based only (mobile-responsive)  
 Secure handling of user data and transactions (SSL/TLS)  
 Compliant with GDPR and PCI-DSS (if handling payments)  
 payment gateway

domain   
 server   
 http- hypertext transfer protocol

3. Functional Requirements  
 3.1 User Registration and Authentication  
 Users can sign up with email and password.  
 Users can log in, log out, and reset passwords.  
 Session/token-based authentication.  
 3.2 Product Management  
 Admins can create, edit, or delete products.  
 Each product has: name, description, price, stock, category, images.  
 3.3 Search and Filtering  
 Users can search products by name, category, or tags.  
 Users can sort by price, popularity, and ratings.  
 3.4 Shopping Cart  
 Users can add/remove products.  
 Quantity can be updated.  
 Cart persists for logged-in users.  
 3.5 Checkout and Payment  
 Users can enter shipping details.  
 Multiple payment options (e.g., credit card, PayPal).  
 Secure payment gateway integration.  
 3.6 Order Management  
 Users can view their past orders.  
 Admins can update order status (Pending, Shipped, Delivered).  
 3.7 Reviews and Ratings  
 Users can rate and review purchased products.  
 4. Non-Functional Requirements  
 4.1 Performance  
 Pages load within 2 seconds.  
 Scalable to support 1000+ concurrent users.  
 4.2 Security  
 All data transferred over HTTPS.  
 Passwords hashed and stored securely.  
 Admin access restricted and logged.  
 4.3 Usability  
 Responsive design for mobile and desktop.  
 Accessible design (WCAG 2.1 compliance).  
 4.4 Maintainability  
 Modular codebase.  
 Proper documentation and code comments.  
 4.5 Reliability  
 99.9% uptime guarantee.  
 Automatic backups and error recovery.  
 5. Future Enhancements  
 Wishlist functionality  
 Real-time chat support  
 - AI-powered product recommendations

- Progressive Web App (PWA) support

5) What is oops

- oops concept   
 class, object, inheritance, polymorphism, abstraction, encapsulation

6) What is class

* class   
   - collection of method, object, datatypes and constructor and variables

7) What is object  
 object - data store - multiple data store   
 object is one kind of variable which store multiple data // class   
 variable - data store at a time single store

8) What is encapsulation  
 encapsulation - data wrap - class - method - data

9) What is inheritance  
 Inheritance - child class can use the functionality of parent class using extends keyword

Advantages  
 - code reusability   
 - code optimization   
 Types of inheritance   
 1 single - 1 child and 1 parent   
 2 multiple - 1 child and multiple parent   
 3 multilevel – Using Step by step  
 4 hybrid - combination any two inheritance   
 5 hierarchical - 1 parent and multiple child class child property

10) What is polymorphism

- Polymorphism

Polymorphism is one interface multiple implementation   
 1 -method overloading - 1 class - more than 1 method -name same - data diff (parameter)   
 2 - method overriding - 1 class - more than 1 method - name same and data same

- abstraction   
 - combination of abstract class and abstract method   
 - for sensitive data hide   
 for Ex. Name, email, pass, bank Aadhar which are sensitive data

11) What is Alpha testing?

- Alpha testing is a type of software testing performed to identify bugs before releasing the product to real users or to the public.

- Alpha testing is one of the user acceptance testing.

12) What is beta testing?

Performed by actual users in a real-world environment, usually before the final release.

13) what is 7 key principles? Explain in detail?

1.testing shows the presence of defects

- Testing can reveal defects, but it cannot prove a software is completely defect-free.

2.Exhaustive testing is impossible

Due to the vast number of possible input combinations and conditions, it’s not feasible to test every scenario.

3. Early testing

Starting testing early in the development lifecycle helps identify and fix defects sooner, reducing costs and effort.

4.Defect clustering

A small subset of modules or functionalities often contains the majority of defects.

5. pesticide paradox

Repeating the same tests can become less effective at finding new defects over time, requiring test cases to be regularly reviewed and updated.

6. Testing is context dependent

The testing approach should be tailored to the specific project, system, and context.

7. Absence of errors fallacy

- If a built software is 99% bug-free it does not follow user requirement then it is unusable. It is not only necessary that software is 99% bug-free but it is also mandatory to fulfil all the customers requirements.

14) Difference between Smoke and Sanity?

- Smoke testing -

Smoke testing is conducted early in the software build process to ensure that the core functionalities are working and the build is stable enough for further testing.

- Sanity testing -

Sanity testing is performed later to verify that specific changes have not led to new issues, ensuring the build is stable after modifications.

15) What is Api testing

- Api stands for the application programming interface

- Basically a collection of functions and procedures which allows us to communicate two applications or libraries.

Advantage:-

1.api is language independent

2.use functionality code not access

15) Types of Api Testing

There are two types of api

1.Rest api

2.Soap api

* Rest api:-

-representational state transfer

-free rules regulation by backend developer

-low net speed to open website

-not secure

-Json,xml,html,text

- Soap api:-

-rules regulation follow compulsory

-secure high

-high net seed to open website

-xml

-Link api given by backend developer And data use frontend developer .

1. simple object access protocol- no protocol follow, low band width call ,less secure

Ex. Json, html, text ,xhtml

2. representational state transfer- protocol follow, high band width call, high secure

Ex.xml

Json:-java script object notation

Xml -> Extensible markup language

Api method (query):-

Get > select data put

Post > insert

Put > update all columns

Delete >delete

Patch >update particular columns

16) 4 pillar of software testing

Collection of code we will use for particular task/functionality

Code that work of particular function

1.unit testing

2.integration testing

3.system testing

4.uat (user acceptance testing)

1.unit testing :-

- single functionality testing at a time

2.integration testing :-

- one or more functionality merge test at a time

3.system testing :-

- website testing all functionality test at least 3 times start to end

4.uat :-

- application test by client

- quality assurance certified

17) What is Exploratory Testing?

* Test case changes after testing by tester.

18) What is Monkey Testing?

* Unknown testing
* Monkey testing is a type of software testing in which the tester tests the application or software by providing some random inputs and checking the behavior of the application or the software. It is also observed by seeing whether the application or software crashes on a given input or not.

19) localization testing:-

> website in multiple language by location like states.

> current location :-language set by location.

> country wise multiple language by state.

-globalization testing:-

> uber,amazon,alibaba.

> world wise multiple language by country.

> language,currency.

-mutation testing:-

> changes in website

-gorilla testing:-

> click more times at a time to crash website like 500 times

-adhoc testing:-

>unknown testing like (without project material)

20) static testing dynamic testing:-

* static testing:-

> before website create testing

> documentation without website open testing.

* dynamic testing:-

> after website open testing.

- review:-

> test case checked by senior tester then done final.

* walkthrough:-

> final test case represented in a team.

* inspection:-

> critical project for

> project manager arranged a meeting.

> particular things testcase.

> then discuss with all meeting members.

21) Bug status:-

- new:- Entry of new bug in a bug report it’s called new.

- assigned:- assigned by tester to developer.

- open:- when developers check the status it’s called open status.

- in progress:- work in progress.

- fix:- work is done.

- retest:- retest by tester for more bug is available or not.

- close:- if no bug is found than close.

- reopen:- if a bug is not solved than reopen.

- reject:- bug assignment by tester but bug not except by developer.

- deferred:-it’s not a bug it’s a work of functionality.

22) What is manual testing ?

>testing by human not need of tools.

* disadvantages:-

>time consuming.

>bug chance high.

>performed by non tech / tech person.

>performance (timing of process load page) testing is not possible.

* advantage:-

>not require technical skills.

>not require tools.

>functional testing is possible.

* performance testing:- automation.

1.load testing

2.stress

3.volume

1.load testing:- automation.

Capacity test by tester generates space for users by developer(increase checked).

2.stress:- automation.

Checked by increase and decrease.

3.volume:- depends on the database.

Performance checked by storage (capacity).

23) GUI testing:-

Graphical user interface

1.responsive testing – bootstrap framework.

2.color and font testing

3.hover>outside design when click

4.label testing

24) Database testing :-

Database - collection of data stored in well (proper) structure from

Developer backend

RDBMS – Relational Database Management System

Database of rdbms :-

Mysql,oracl,postgree,sqlite

SQL- Structure Query Language

- use language in database

- no sql

- json (java script object notation)

- mongodb

View : -

View is a virtual table

Original table and virtual table both are connected with each other

QUERY : -

* Create database database name
* Drop database database name
* Create table tablename
* Drop table tablename
* Create view viewname

* For particular things delete from table

-delete from tablename where price = 4

-select \* from tablename

When \* is all column

-select \* from tablename where pid=6

Particular column

-update tablename set pname=”tv”,price=”5000” where pid=3;

* Update and delete only one time at a one query.
* Insert into tablename(empname ,emdesignation,

empaddress)VALUE(“drumil”,”xyz”,”ahmd”);

* Select customer.cname,customer.caddress,

orders.pname,orders.custid from customer

INNER JOIN orders ON customer.cid=orders.custid