1. SDLC (Software Development Life Cycle)

    a. Typical SDLC environments

No	Typical SDLC environments
1.	Development Environtment : It is where prototyping or programming takes place before moving to the test environment. It is a dedicated workplace (e.g. servers, client stations, database, etc) and tools for developing, and debugging an application or program. In this workplace is where the developer tests code and checks whether the application runs successfully with that code. Once the application has been tested and the developer feels that the code is working fine, the application then moves to the testing environment also called an Integrated Development Environment (IDE).
2.	Testing Environtment : The test environment is where formal verification and validation are performed before moving to the operational environment fully implemented for production use. This workplace environment and its configuration must be similar to the operational (production) environment. It is a dedicated workplace (e.g. servers, client stations, database, etc) and tools to make formal testing on applications or programs. Commonly, called the validation environment since testing is done also in the development and operational environments. The application is verified and validated in the testing environment to check for reliability and to make sure it does not fail. A fully documented validation is performed in the testing environment as the final step before the application could be deployed in an operational environment. The validation needs to be approved in order to deploy it in the operational environment.
3.	Operational Environment: Once the application is verified and validated, the application then becomes a part of the operational environment. It is known as the production environment because it is where the application go-lives. Also called the production environment, it is the workplace where the system is in its target environment or final location for commercial or deployment use. The operational environment is a workplace where software and other products are actually challenged once they are placed into operation for their intended uses by end-users. An operational environment contains the real-time setting where programs are run and hardware setups are installed and relied on for organization or commercial daily operations.

b. Tescase development :
   - Waterfall Method

The fourth stage, entered in the process of integration and system testing. At this stage, the modules that have been created in the previous stage will be combined. After the system integration process has been completed, the next step is to enter the module testing.Which aims to find out whether the software is in accordance with the design, and the functionality of the application is running well or not. So, with the testing stage, it can prevent errors, bugs, or errors in the program before entering the production stage

   - Agile Method

testing practice that follows the rules and principles of agile software development. Unlike the Waterfall method, Agile Testing can begin at the start of the project with continuous integration between development and testing. Agile Testing methodology is not sequential (in the sense it's executed only after coding phase) but continuous.

2. All about tests basics understanding

   a. This type of testing is performed before the actual system testing to check if the critical functionalities are working fine in order to carry out further extensive testing

   b. Regression testing is carried out upon receiving the build of the software after fixing the bugs that were found in the initial round of testing. It verifies whether the bug is fixed and checks if the entire software is working fine with the changes.

c. The Black Box Test is a test that only considers the external behavior of the system; the internal workings of the software is not taken into account. The White Box Test is a method used to test a software taking into consideration its internal functioning. It is carried out by testers.

d. - Unit testing : Unit testing is the first testing phase and it is practiced before moving to the phase of integration testing.

- Integration Testing is performed when two or more functions or components of the software are integrated to form a system. It basically checks the proper functioning of the software when the components are merged to work as a single unit.

e. - Functional testing verifies that the operational execution of a program or mobile app happens according to the technical and business requirements. Only if every feature of a software system works correctly, it can pass a functional test. Functional testing is usually conducted before non-functional testing and is done manually. The tester provides specific inputs to the program and compares the result with the expected output. Functional testers are not concerned about the source code but focus on checking the functionality.

- Non-functional testing is testing where the tester will test the non-functional parameters, for example, performance, reliability, load test, and accountability of the software or application.

f. - Component testing is module testing, is done after unit testing. In this type of testing those test objects can be tested independently as a component without integrating with other components e.g. modules, classes, objects, and programs. This testing is done by the development team.

- Integration Testing is performed when two or more functions or components of the software are integrated to form a system. It basically checks the proper functioning of the software when the components are merged to work as a single unit.

g. - Accessiblility Testing : practice of ensuring your mobile and web apps are working and usable for users without and with disabilities such as vision impairment, hearing disabilities, and other physical or cognitive conditions.

- Acceptance Testing : ensures that the end-user (customers) can achieve the goals set in the business requirements, which determines whether the software is acceptable for delivery or not. It is also known as user acceptance testing (UAT).

- Black box Testing : involves testing against a system where the code and paths are invisible.

- End to end Testing : technique that tests the application’s workflow from beginning to end to make sure everything functions as expected.

- Functional Testing : Functional testing checks an application, website, or system ensure it’s doing exactly what it’s supposed to be doing.

- Interactive Testing : enables testers to create and facilitate manual tests for those who do not use automation and collect results from external tests.

- Integration Testing : Integration testing ensures that an entire, integrated system meets a set of requirements. It is performed in an integrated hardware and software environment to ensure that the entire system functions properly.

- Load Testing : This type of non-functional software testing process determines how the software application behaves while being accessed by multiple users simultaneously.

- Non Functional Testing : verifies the readiness of a system according to nonfunctional parameters (performance, accessibility, UX, etc.)

- Performance Testing : Performance testing examines the speed, stability, reliability, scalability, and resource usage of a software application under a specified workload.

- Regression Testing : Regression testing is performed to determine if code modifications break an application or consume resources.

- Sanity Testing : Performed after bug fixes, sanity testing determines that the bugs are fixed and that no further issues are introduced to these changes.

- Security Testing : Security testing unveils the vulnerabilities of the system to ensure that the software system and application are free from any threats or risks.

- Single user Performance Testing : checks that the application under test performs fine according to specified threshold without any system load.

- Smoke Testing : validates the stability of a software application, it is performed on the initial software build to ensure that the critical functions of the program are working.

- Stress Testing : activity that tests beyond normal operational capacity to test the results.

- Unit Testing : process of checking small pieces of code to ensure that the individual parts of a program work properly on their own, speeding up testing strategies and reducing wasted tests.

h. Step by step flow of doing testing :

1. Define the scope : First, consider what the testing process aims to achieve, and run a feasibility analysis. Which of the tests can be automated and which require manual involvement

2. Choose an automation tool : choose a tool that suits your requirements, but the choice will also depend on the technology of the web application being tested. There are many automation tools to choose from, consider cost, functionality, intuition and flexibility.

3. Design a strategy : The QA team should create a test plan that sets out the approach and end-goal of the project, and choose a suitable framework for the test cases to operate in. The framework should contain common practices, testing tools, and standards. Common test automation frameworks include data-driven, keyword-driven, linear scripting, and modular testing.

4. Set the environment : creating the right environment for testing, and maximizing test coverage across multiple scenarios.

5. Write a script : write scripts to perform the tests, based on actual requirements and scripting standards.

6. Execute the test : Ada dua cara utama untuk mengotomatisasi pengujian QA: pengujian GUI (antarmuka pengguna grafis) dan pengujian API. GUI adalah jenis pengujian QA yang bertujuan untuk meniru pengalaman pengguna dengan perangkat lunak seperti TestComplete. Itu dapat dijalankan dengan cara yang sama setiap kali, atau menguji fitur perangkat lunak yang berbeda di setiap iterasi.

7. Analyze and report : Once the tests have been executed, the automated tool will generate a report. The results will show which components contain bugs or defects, and whether or not additional testing is required.

3. About HTTP Request

a. - GET method is used to retrieve information from the given server using a given URI. Requests using GET should only retrieve data and should have no other effect on the data.

- POST request is used to send data to the server, for example, customer information, file upload, etc. using HTML forms

b. - GET request is used to read/retrieve data from a web server. GET returns an HTTP status code of 200 (OK) if the data is successfully retrieved from the server

- POST request is used to send data (file, form data, etc.) to the server. On successful creation, it returns an HTTP status code of 201.

- PUT request is used to modify the data on the server. It replaces the entire content at a particular location with data that is passed in the body payload. If there are no resources that match the request, it will generate one.

- PATCH is similar to PUT request, but the only difference is, it modifies a part of the data. It will only replace the content that you want to update.

- DELETE request is used to delete the data on the server at a specified location.

c. URL parameters are dynamic elements of a query string. A query string is the part of the URL that follows a question mark symbol (?).

d. The Content-Type header is used to indicate the media type of the resource. The media type is a string sent along with the file indicating the format of the file. For example, for image file its media type will be like image/png or image/jpg, etc. In response, it tells about the type of returned content, to the client.

e. Header : Authorization Used when the client desires to authenticate the request. Credentials containing authenticate information are sent to the server (for example, "Authorization: Basic AbCdE123$5").

f. status code 100 : (Continue), status code 101 : (Switching), status code 102 : (Processing), status code 200: (Ok), status code 201: (Created), status code 204: (No Content), status code 300: (Multiple choice), status code 302: (Found), status code 303: (See Other), status code 400: (Bad Request), status code 401: (Unauthorized), status code 404: (Not found), status code 500: (Internal server error), status code 502: (Bad gateway), status code 503: (Service unavailable).

g. check for any running Http request in the background on a website:

1. Open the webpage whose headers have to be checked.
2. Right click and select 'Inspect' to open developer tools.
3. Select the network tab and refresh or reload the page.
4. Select any HTTP request from the left panel and the header will be displayed on the right.

4. Bugs

a. Bug is a feature is flawed or a gap in the system that can cause a program or application to not run properly

b. Using Agile Methodology, Agile software development methodology is a project management technique that prioritizes continuous development systems. That is, the objectives of each development phase will differ from the previous phase depending on the changes required.

c. If the bug is not reproducible, then the testing effort used in finding, analyzing and reporting that particular bug/defect is a total waste.For understanding and reproducing a bug, it is essential to have detailed and properly explained Steps to Reproduce, state and environment in which the bug occurred.

5. elements and functionality that shall be present on a sign up and signin
   page : phone number, username, email, password, button login, forgot password, sign facebook, sign google, button register, full name, first name, last name, gender, date of birth

6. In exel.