1. **What is usability testing in web testing?**

   Ans. Usability Testing is a technique used to evaluate a product (in this case a website) by testing is on users. Most people who set up a usability test carefully construct a scenario wherein a person performs a list of tasks that someone who is using the website for the first time is likely to perform.

2. **Explain the difference between HTTP and HTTPS?**

   Ans. HyperText Transfer Protocol Secure (HTTPS) is the secure version of HTTP, the protocol over which data is sent between your browser and the website that you are connected to. The 'S' at the end of HTTPS stands for 'Secure'. It means all communications between your browser and the website are encrypted.

3. **Write the test scenarios for testing a web site?**
   - All mandatory fields should be validated and indicated by an asterisk (*) symbol.
   - Validation error messages should be displayed properly at a correct position.
   - All error messages should be displayed in the same CSS style (e.g. using red color)
   - General confirmation messages should be displayed using CSS style other than error messages style (e.g. using green color)
   - Tooltips text should be meaningful.
   - Drop-down fields should have the first entry as blank or text like ‘Select’.
   - Delete functionality’ for any record on a page should ask for a confirmation.
   - Select/deselect all records option should be provided if page supports record add/delete/update functionality
   - Amount values should be displayed with correct currency symbols.
   - Default page sorting should be provided.
   - Reset button functionality should set default values for all fields.
   - All numeric values should be formatted properly.
   - Input fields should be checked for the max field value. Input values greater than specified max limit should not be accepted or stored in the database.
   - Check all input fields for special characters.

4. **Write a few Test Cases on GMail functionality.**

   Test Scenarios for Inbox Functionality(Receive Email)

   - Verify that a newly received email is displayed as highlighted in the Inbox section.
   - Verify that a newly received email has correctly displayed sender emailId or name, mail subject and mail body(trimmed to single line).
   - Verify that on clicking the newly received email, user is navigated to email content.
   - Verify that the email contents are correctly displayed with the desired source formatting.
   - Verify that any attachments are attached to the email and is downloadable.

- Verify that the attachments are scanned for viruses before download.
- Verify that all the emails marked as read are not highlighted.
- Verify that all the emails read as well as unread have a mail read time appended at the end on the email list displayed in the inbox section.
- Verify that count of unread emails is displayed alongside 'Inbox' text in left sidebar of GMail.
- Verify that unread email count increases by one on receiving a new email.
- Verify that unread email count decreases by one on reading an email (marking email as read).
- Verify that email recipients in cc are visible to all user.
- Verify that email recipients in bcc are not visible to user.
- Verify that all received emails get piled up in the 'Inbox' section and gets deleted in cyclic fashion based on the size availability.
- Verify that email can be received from non-gmail emailIds like - yahoo, hotmail etc.

Test scenarios for Compose mail Functionality

- Verify that on clicking 'Compose' button, a frame to compose a mail gets displayed.
- Verify that user can enter emailIds in 'To', 'cc' and 'bcc' sections and also user will get suggestions while typing the emailds based on the existing emailIds in user's email list.
- Verify that user can enter multiple comma separated emailIds in 'To', 'cc' and 'bcc' sections.
- Verify that user can type Subject line in the 'Subject' textbox.
- Verify that user can type the email in email-body section.

5. **Write any 5 common ATM Machine functionality.**
    - Activation of debit card
    - Withdrawals
    - Deposits
    - Balance Inquiry
    - Change PIN

6. **Give some examples of web applications that are used in our day to day life.**
    - GMAIL
    - Facebook
    - Tutorialspoint
    - Wikipedia
    - Google Docs
    - Paytm
    - Myntra

7. **What are the advantages of Using Cookies?**

Ans. Cookies are a powerful tool because they allow web developers to easily perform long-term user recognition. A Web server has no memory so the hosted Web site you are visiting transfers a **cookie** file of the browser on your computer's hard disk so that the Web site can remember who you are and your preferences. This message exchange allows the Web server to use this information to present you with customized Web pages.

8. **What is XSS and how We can prevent it?**

Ans. Cross-site Scripting (XSS) attacks occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script, to a different end user. Flaws that allow these attacks to succeed are quite widespread and occur anywhere a web application uses input from a user in the output it generates without validating or encoding it.

Prevention for XSS:

1. **Escaping**

The first method you can and should use to prevent XSS vulnerabilities from appearing in your applications is by escaping user input. Escaping data means taking the data an application has received and ensuring it's secure before rendering it for the end user. By escaping user input, key characters in the data received by a web page will be prevented from being interpreted in any malicious way. In essence, you're censoring the data your web page receives in a way that will disallow the characters – especially < and > characters – from being rendered, which otherwise could cause harm to the application and/or users.

2. **Validating Input**

As Troy Hunt so eloquently puts it: "The theory goes like this: Expect any untrusted data to be malicious. What's untrusted data? Anything that originates from outside the system and you don't have absolute control over so that includes form data, query strings, cookies, other request headers, data from other systems (i.e. from web services) and basically anything that you can't be 100% confident doesn't contain evil things."

Validating input is the process of ensuring an application is rendering the correct data and preventing malicious data from doing harm to the site, database, and users. While whitelisting and input validation are more commonly associated with SQL injection

3. **Sanitizing**

A third way to prevent cross-site scripting attacks is to sanitize user input. Sanitizing data is a strong defense, but should not be used alone to battle XSS attacks. It's totally possible you'll find the need to use all three methods of prevention in working towards a more secure application. Sanitizing user input is especially helpful on sites that allow HTML markup, to ensure data received can do no harm to users as well as your database by scrubbing the data clean of potentially harmful markup, changing unacceptable user input to an acceptable format.

9. **Write a few Cross Browsing Testing TCs for any website.**
    - Does the website loads on browser?
    - Does the elements (such as buttons, forms, menu) visible?
    - Does this website or app opens on tablet?
    - Does this website opens on smartphone?
    - Does the dynamic data appears properly in the responsive layout?
    - Does the tables render properly for viewing on specific resolution?
    - Does the data appears correctly in the respective tables?
    - Does the website loads partially under slow connection?