**1. What is bad design?**

* Bad design is when something doesn’t work well, looks unattractive, or is hard to use. It can happen due to poor choices or not testing things properly.

**2. Types of design**

* There are many types of design, including:
  + Graphic design (like posters)
  + User experience (UX) design (how people feel using something)
  + User interface (UI) design (how something looks on a screen)
  + Industrial design (physical products)
  + Interior design (inside spaces)
  + Fashion design (clothes)
  + Patterns of design: Good Design and bad design

**3. What is Jugad?**

* Jugad is a Hindi word for a clever fix or a quick solution to a problem, especially when resources are limited.

**4. What is Feedback?**

* Feedback is the response you get after doing something, helping you understand if you did it right or wrong.

**5. What is a constraint?**

* A constraint is a limit or rule that affects how something can be designed or built.

**6. What is the need for Feedback and Constraints?**

* Feedback helps users know what they did well or poorly, while constraints help designers focus on what’s important and make better decisions.

**7. Explain Feedback and Constraints with example**

* For example, in a flight booking app:
  + **Feedback:** When you enter travel dates, the app shows available flights.
  + **Constraints:** The app must fit on a small phone screen and prioritize key features.

**8. Examples where feedback is needed but not present**

* A website that doesn’t show a message after you submit a form.
* An app that doesn’t confirm when you make a payment.
* A product that doesn’t show if it’s turned on or off.

**9. Explain products that give feedback and have some constraints**

* A smartwatch that tracks fitness (gives feedback) but has a small battery (constraint).
* A video game that shows your score (feedback) but has limited graphics (constraint).
* A website that tells you about your account (feedback) but has to keep your data safe (constraint).

**10. What is Prototype? Explain with example**

* A prototype is a simple version of a design used to test ideas. For example, a paper version of a mobile app to see how it might look and work.

**11. What is Wireframe?**

* A wireframe is a basic sketch of how a website or app will look, showing where things will go.

**12. Explain types of users**

* Types of users include:
  + **Primary users:** The main people using a product.
  + **Secondary users:** People who use it indirectly.
  + **Tertiary users:** People affected by it but don’t use it directly.

**13. What is High-Fidelity and Low-Fidelity Designs?**

* High-fidelity designs are detailed and close to the final product. Low-fidelity designs are simple and rough sketches.

**14. What is the need for wireframes? Explain with example**

* Wireframes help plan how a product will look and work. For example, a wireframe for a website shows where the menu and content will go.

**15. What is CSS? Explain types of CSS**

* CSS (Cascading Style Sheets) is a language used to style websites. Types of CSS include:
  + **Inline CSS:** Styles applied directly in HTML tags.
  + **Internal CSS:** Styles written in the head of an HTML document.
  + **External CSS:** Styles written in a separate file linked to the HTML.

**16. Explain CSS Syntax**

* CSS syntax is made up of selectors, properties, and values. For example: **h1 { color: blue; }** means "make all h1 headings blue".

**17. Explain CSS Box Model**

* The CSS box model is how CSS lays out elements on a page. It includes the content area, padding, border, and margin.

**18. What are the types of writing CSS?**

* There are three main types:
  + **Inline CSS:** Written directly in HTML tags.
  + **Internal CSS:** Written in the head of an HTML document.
  + **External CSS:** Written in a separate file linked to the HTML.

**19. What are various CSS Properties?**

* CSS properties control the layout and appearance of elements. Examples include:
  + **color** for text color
  + **background-color** for background color
  + **font-size** for text size
  + **margin** for space around elements

**20. What is CMS? Explain advantages**

* A CMS (Content Management System) is software that helps create and manage digital content. Advantages include:
  + Easy to update content
  + No need to know how to code
  + Can be used by multiple people
  + Secure and scalable

**21. Advantages of Content Management Systems**

* Advantages include:
  + Easy to use
  + Cost-effective
  + Scalable
  + Secure
  + Can be customized

**22. List CMS Development tools**

* Some popular CMS development tools include:
  + WordPress
  + Joomla
  + Drupal
  + Magento
  + Shopify

**23. How WordPress is used for creation of interactive website?**

* WordPress can be used to create interactive websites by:
  + Installing plugins for interactive features
  + Using themes designed for interactive websites
  + Creating custom code for specific interactions

**24. What is Website Deployment?**

* Website deployment is the process of making a website live and accessible to the public.

**25. What are the steps to deploy a website on web server?**

* Steps include:
  + Uploading files to the server
  + Configuring server settings
  + Setting up databases
  + Testing the website

**26. Explain concept of Parking a website**

* Parking a website means temporarily hosting a website on a server while the final website is being developed.

**27. List and explain Nielsen's heuristics Rules**

* Nielsen's heuristics are 10 rules for designing user-friendly interfaces:
  1. Visibility of system status
  2. Match between system and the real world
  3. User control and freedom
  4. Consistency and standards
  5. Error prevention
  6. Recognition rather than recall
  7. Flexibility and efficiency of use
  8. Aesthetic and minimalist design
  9. Help users recognize, diagnose, and recover from errors
  10. Help and documentation

**28. Evaluate any UI / product using Ben Shneiderman’s eight golden rules for interface design**

* Shneiderman's eight golden rules are:
  1. Strive for consistency
  2. Enable frequent users to use shortcuts
  3. Offer informative feedback
  4. Design dialogue to yield closure
  5. Offer simple error handling
  6. Permit easy reversal of actions
  7. Support internal locus of control
  8. Reduce short-term memory load

**29. How Evaluation methods are evaluated**

* Evaluation methods are evaluated based on their effectiveness, efficiency, and usability.

**30. Designing an Evaluation Strategy**

* An evaluation strategy involves:
  + Defining goals and objectives
  + Identifying target audience
  + Selecting evaluation methods
  + Collecting and analyzing data
  + Reporting results

**Purpose of Evaluation**

* The purpose of evaluation is to:
  + Assess the effectiveness of a design
  + Identify areas for improvement
  + Inform design decisions
  + Measure user satisfaction
  + Improve overall user experience
* -----------------------------------------------------------------------

**Explanation of Key Sections and Tags (For this expl. Refer Task 8 – project 1 in google drive)**

1. **<!DOCTYPE html>**
   * This declaration tells the browser that the document is an HTML5 document. It helps the browser render the page correctly.
2. **<html lang="en">**
   * This tag starts the HTML document and specifies that the language of the document is English (**lang="en"**).
3. **<head>**
   * This section contains meta-information about the document, such as the character set, viewport settings for responsive design, the title of the page, and links to stylesheets.
4. **<meta charset="UTF-8">**
   * This tag specifies the character encoding for the document, allowing it to display a wide range of characters.
5. **<meta name="viewport" content="width=device-width, initial-scale=1.0">**
   * This tag makes the webpage responsive, ensuring it looks good on all devices by setting the width to the device's width.
6. **<title>**
   * This tag sets the title of the webpage that appears in the browser tab.
7. **<link rel="stylesheet" href="clg.css">**
   * This tag links an external CSS file (**clg.css**) that styles the webpage.
8. **<body>**
   * This tag contains all the content that is visible on the webpage.

**Header and Navigation**

1. **<header>**
   * This tag defines the header section of the webpage, which typically contains the logo and navigation links.
2. **<div class="container">**
   * A **div** tag with a class of "container" is used to group elements together and apply styles.
3. **<div class="logo">**
   * This **div** contains the college logo image.
4. **<ul> and <li>**
   * The **<ul>** tag creates an unordered list. Inside it, **<li>** tags represent individual list items, which in this case are navigation links.
5. **<nav>**
   * This tag defines the navigation section of the page, containing links to different sections of the website.

**Hero Section**

1. **<section id="home" class="hero">**
   * This tag defines a section of the webpage. The **id** allows for linking to this section, and the **class** is used for styling.
2. **<h1> and <p>**
   * The **<h1>** tag is used for the main heading, and **<p>** is for a paragraph of text.
3. **<a>**
   * This tag creates a hyperlink. In this case, it links to the admissions section with the text "Apply Now."

**About Section**

1. **<section id="about" class="about">**
   * Another section, this one for information about the college.
2. **<img>**
   * This tag displays an image. The **src** attribute specifies the image URL, and **alt** provides alternative text for the image.

**Academics Section**

1. **<section id="academics" class="academics">**
   * This section provides information about academic programs.
2. **<div class="card">**
   * Each card represents a program and is styled as a separate block.

**Placements Section**

1. **<section id="placements" class="placements">**
   * This section highlights placement opportunities for graduates.

**Footer**

1. **<footer>**
   * This tag defines the footer of the webpage, which usually contains copyright information and links.
2. **<p>**
   * The paragraph tag is used to display text, such as copyright information.
3. **<div class="social-links">**
   * This **div** contains links to social media platforms.

**CSS Styling**

* The CSS styling is included within the **<style>** tag in the hero section, which sets a background image for that section.

**Summary**

* The HTML code creates a structured webpage with various sections, each serving a specific purpose (header, hero, about, academics, placements, and footer).
* Tags like **<header>**, **<section>**, **<nav>**, **<ul>**, **<li>**, and **<footer>** help organize the content logically.
* The use of **<img>** for images and **<a>** for links enhances the interactivity

**Overview of the CSS Code**

This CSS code styles the webpage for "Zeal College of Engineering & Research." It defines how different elements on the webpage should look, including layout, colors, fonts, and spacing.

**Explanation of Key Sections and Rules**

1. **Universal Selector (\*)**

css

VerifyOpen In EditorEditCopy code

1\* {

2 margin: 0;

3 padding: 0;

4 box-sizing: border-box;

5 font-family: 'Arial', sans-serif;

6}

* + **Purpose:** Resets default margin and padding for all elements to avoid unwanted spacing.
  + **box-sizing: border-box;**: Ensures that padding and borders are included in the element's total width and height.
  + **font-family: 'Arial', sans-serif;**: Sets the default font for the entire page to Arial.

1. **Body Styles**

css

VerifyOpen In EditorEditCopy code

1body {

2 line-height: 1.6;

3 background-color: #eef2f3;

4}

* + **Purpose:** Sets the line height for readability and the background color for the body of the page.

1. **Container Class**

css

VerifyOpen In EditorEditCopy code

1.container {

2 width: 85%;

3 margin: 0 auto;

4}

* + **Purpose:** Centers the content and limits its width to 85% of the viewport.

**Header and Navigation Styles**

1. **Header Styles**

css

VerifyOpen In EditorEditCopy code

1header {

2 background-color: #002244;

3 color: #ffffff;

4 padding: 25px 0;

5}

* + **Purpose:** Sets the background color, text color, and padding for the header section.

1. **Logo Image**

css

VerifyOpen In EditorEditCopy code

1header .logo img {

2 width: 130px;

3}

* + **Purpose:** Sets a fixed width for the logo image.

1. **Navigation Styles**

css

VerifyOpen In EditorEditCopy code

1nav ul {

2 display: flex;

3 justify-content: space-evenly;

4 list-style: none;

5}

* + **Purpose:** Displays the navigation items in a horizontal row using Flexbox and removes bullet points.

1. **Navigation Links**

css

VerifyOpen In EditorEditCopy code

1nav ul li a {

2 color: #ffffff;

3 text-decoration: none;

4 padding: 12px 25px;

5 transition: background-color 0.3s ease;

6}

* + **Purpose:** Styles the links in the navigation bar, including color, padding, and a transition effect for hover.

1. **Hover Effect for Links**

css

VerifyOpen In EditorEditCopy code

1nav ul li a:hover {

2 background-color: #ffc107;

3 border-radius: 5px;

4}

* + **Purpose:** Changes the background color and adds rounded corners when a user hovers over a link.

**Hero Section Styles**

1. **Hero Section**

css

VerifyOpen In EditorEditCopy code

1.hero {

2 background-size: cover;

3 height: 450px;

4 display: flex;

5 align-items: center;

6 justify-content: center;

7 text-align: center;

8 color: white;

9}

* + **Purpose:** Sets the hero section's height, centers its content, and applies a white text color.

1. **Hero Heading and Paragraph**

css

VerifyOpen In EditorEditCopy code

1.hero h1 {

2 font-size: 3.5rem;

3 margin-bottom: 15px;

4}

5

6.hero p {

7 font-size: 1.6rem;

8 margin-bottom: 25px;

9}

* + **Purpose:** Styles the main heading and paragraph within the hero section, adjusting font sizes and margins.

1. **Button Styles**

css

VerifyOpen In EditorEditCopy code

1.btn {

2 background-color: #ffc107;

3 padding: 12px 25px;

4 color: #002244;

5 text-decoration: none;

6 font-weight: bold;

7 border-radius: 5px;

8}

9

10.btn:hover {

11 background-color: #e6a800;

12}

* + **Purpose:** Styles the button, including background color, padding, text color, and hover effect.

**About Section Styles**

1. **About Section**

css

VerifyOpen In EditorEditCopy code

1.about {

2 padding: 60px 0;

3 background-color: #ffffff;

4 text-align: center;

5}

* + **Purpose:** Adds padding and sets the background color for the About section.

1. **About Heading and Image**

css

VerifyOpen In EditorEditCopy code

1.about h2 {

2 font-size: 2.8rem;

3 margin-bottom: 25px;

4}

5

6.about-image {

7 width: 650px;

8 height: auto;

9 margin-top: 25px;

Html and inline css:  
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Interview Dry Run Test</title>

</head>

<body style="font-family: Arial, sans-serif; margin: 20px; background-color: #f4f4f4;">

<h2 style="color: #333;">Interview Dry Run Test</h2>

<form style="background: #fff; padding: 20px; border-radius: 5px; box-shadow: 0 2px 5px rgba(0,0,0,0.1);">

<label for="candidateName" style="display: block; margin-bottom: 5px;">Candidate Name:</label>

<input type="text" id="candidateName" name="candidateName" style="width: 100%; padding: 10px; margin-bottom: 10px; border: 1px solid #ccc; border-radius: 4px;" required>

<label for="interviewDate" style="display: block; margin-bottom: 5px;">Interview Date:</label>

<input type="date" id="interviewDate" name="interviewDate" style="width: 100%; padding: 10px; margin-bottom: 10px; border: 1px solid #ccc; border-radius: 4px;" required>

<label for="comments" style="display: block; margin-bottom: 5px;">Comments:</label>

<textarea id="comments" name="comments" rows="4" style="width: 100%; padding: 10px; margin-bottom: 10px; border: 1px solid #ccc; border-radius: 4px;"></textarea>

<button type="submit" style="background-color: #28a745; color: white; padding: 10px 15px; border: none; border-radius: 4px; cursor: pointer;">Submit</button>

</form>

</body>

</html>

-----------------------------

Internal css:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Interview Dry Run Test</title>

<style>

body { font-family: Arial, sans-serif; margin: 20px; background: #f4f4f4; }

form { background: #fff; padding: 20px; border-radius: 5px; box-shadow: 0 2px 5px rgba(0,0,0,0.1); }

label { display: block; margin-bottom: 5px; }

input, textarea { width: 100%; padding: 10px; margin-bottom: 10px; border: 1px solid #ccc; border-radius: 4px; }

button { background: #28a745; color: white; padding: 10px 15px; border: none; border-radius: 4px; cursor: pointer; }

</style>

</head>

<body>

<h2>Interview Dry Run Test</h2>

<form>

<label for="candidateName">Candidate Name:</label>

<input type="text" id="candidateName" name="candidateName" required>

<label for="interviewDate">Interview Date:</label>

<input type="date" id="interviewDate" name="interviewDate" required>

<label for="comments">Comments:</label>

<textarea id="comments" name="comments" rows="4"></textarea>

<button type="submit">Submit</button>

</form>

</body>

</html>

External css:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Interview Dry Run Test</title>

<link rel="stylesheet" href="styles.css">

</head>

<body>

<h2>Interview Dry Run Test</h2>

<form>

<label>Candidate Name:</label>

<input type="text" required>

<label>Interview Date:</label>

<input type="date" required>

<label>Comments:</label>

<textarea rows="4"></textarea>

<button type="submit">Submit</button>

</form>

</body>

</html>

Css:

body { font-family: Arial, sans-serif; margin: 20px; background: #f4f4f4; }

form { background: #fff; padding: 20px; border-radius: 5px; box-shadow: 0 2px 5px rgba(0,0,0,0.1); }

label { display: block; margin: 5px 0; }

input, textarea { width: 100%; padding: 10px; margin: 10px 0; border: 1px solid #ccc; border-radius: 4px; }

button { background: #28a745; color: white; padding: 10px 15px; border: none; border-radius: 4px; cursor: pointer; }

1. **Inline CSS**:
   * Styles are applied directly within HTML elements using the **style** attribute.
   * Example: **<h1 style="color: blue;">Hello World</h1>**
2. **Internal CSS**:
   * Styles are defined within a **<style>** tag in the **<head>** section of an HTML document.
   * Example:

html

EditRunCopy code

1<style>

2 h1 { color: blue; }

3</style>

1. **External CSS**:
   * Styles are written in a separate **.css** file, which is linked to the HTML document using the **<link>** tag.
   * Example:

html

EditRunCopy code

1<link rel="stylesheet" href="styles.css">

**what are the principles of hci:**

1. Usability:
   * Systems should be easy to use and understand, minimizing the learning curve for users.
2. User -Centered Design:
   * Design should be based on the needs, preferences, and limitations of the end-users. Involves user feedback and testing throughout the design process.
3. Consistency:
   * Similar tasks should be performed in similar ways, ensuring a predictable experience across the system.
4. Feedback:
   * Users should receive clear and immediate feedback about their actions, including confirmations, errors, and status updates.
5. Accessibility:
   * Interfaces should be designed to be usable by people with a wide range of abilities and disabilities, ensuring inclusivity.
6. Error Prevention and Recovery:
   * Systems should help prevent errors and provide easy ways for users to recover from mistakes.
7. Affordance:
   * Design elements should suggest their usage; for example, buttons should look clickable.
8. Simplicity:
   * The interface should be as simple as possible, avoiding unnecessary complexity and clutter.
9. Flexibility:
   * Users should have the option to tailor the interface to their personal preferences and workflows.
10. Context Awareness:
    * Systems should consider the context in which they are used, adapting to different environments and user situations.

**What is Design?**

It involves problem-solving and creativity to develop solutions that meet specific needs or goals, often focusing on aesthetics, functionality, and user experience.

Golden Rules for Design:

1. Consistency:
   * Maintain uniformity in design elements to create a cohesive experience.
2. Feedback:
   * Provide users with clear, immediate responses to their actions.
3. Simplicity:
   * Keep the design straightforward and uncluttered to enhance usability.
4. Visibility:
   * Ensure important features are easily discoverable and accessible.
5. User Control:
   * Allow users to feel in control of their interactions and provide options for undoing actions.
6. Error Prevention:
   * Design to minimize the likelihood of user errors and provide clear guidance when errors occur.

**MVC Framework**

**MVC (Model-View-Controller)** is a software architectural pattern used for developing user interfaces by dividing an application into three interconnected components:

1. **Model**:
   * Represents the data and business logic of the application. It manages the data, logic, and rules of the application.
2. **View**:
   * Represents the user interface and displays data to the user. It presents the model's data to the user and sends user commands to the controller.
3. **Controller**:
   * Acts as an intermediary between the Model and View. It processes user input, interacts with the model, and updates the view accordingly.

**Key Benefits:**

* **Separation of Concerns**: Each component has a distinct responsibility, making the application easier to manage and scale.
* **Reusability**: Components can be reused across different parts of the application.
* **Testability**: Each component can be tested independently, improving overall code quality.

Norman's Seven Principles of Design

focus on usability and user experience. Here they are in very short form:

1. **Visibility**: Ensure important functions are visible and easily accessible.
2. **Feedback**: Provide clear feedback to users about their actions and system status.
3. **Constraints**: Limit user actions to prevent errors and guide them toward correct usage.
4. **Mapping**: Design controls that logically relate to their effects, making them intuitive.
5. **Consistency**: Maintain uniformity in design elements and behavior across the interface.
6. **Affordance**: Design elements should suggest their functionality (e.g., buttons should look clickable).
7. **Error Prevention**: Design to minimize the chance of errors and provide helpful error messages.

Examples:

**Good Design**

A smartphone app with a clean interface, intuitive navigation, and accessible features that enhance user experience.

**Bad Design**

A website with cluttered text, confusing navigation, and poor color contrast that makes it difficult for users to find information.

**Feedback and Constraint**

Feedback: A form validation message that alerts users when they enter an invalid email address. Constraint: Limiting password length to 8-16 characters to enhance security.

**Prototype**

A clickable mockup of a mobile app that simulates user interactions to test functionality and design before development.

**Wireframe**

A basic layout sketch of a webpage showing the placement of elements like headers, images, and buttons without detailed design elements.