

Ui Ux Design Assignments

Introduction to Ui/Ux Design

Theory Assignments:

1. **Essay:** Define UI and UX and explain their differences. Provide real-life examples where UI and UX impact user behavior.
2. **Research Task:** Identify and analyze a case study of a popular app or website. Discuss how UI and UX principles are applied to enhance user engagement.
3. **Reflection:** Write about a personal experience with a poorly designed website/app and explain how the UX could be improved.

Practical Assignments:

1. **Design Analysis:** Review a popular website (e.g., Apple or Airbnb) and identify key UI/UX elements.
2. **Design Critique:** Find an example of a poorly designed app and document specific UI and UX improvements.
3. **Comparison Task:** Compare the UI and UX of two similar apps, noting strengths and weaknesses.

Understanding Users and Research Techniques

Theory Assignments:

1. **Research Paper:** Discuss the importance of user research in UI/UX design. Include methods and best practices.
2. **Concept Understanding:** Write about empathy mapping and how it helps in creating a user-centered design.

Practical Assignments:

1. **Create Personas:** Conduct a mini-survey, gather responses, and create user personas based on the findings.
2. **User Journey Mapping:** Map out a user journey for a hypothetical product, detailing pain points and opportunities for improvement.
3. **Competitive Analysis:** Choose two competing products and conduct a comparative analysis focusing on user experience and design.

Fundamentals of Visual Design

Theory Assignments:

1. **Study on Design Principles:** Explain principles like contrast, balance, and hierarchy with examples.
2. **Color Theory:** Write a report on color psychology and its impact on design.

Practical Assignments:

1. **Mood Board Creation:** Create a mood board for a hypothetical brand or app, focusing on visual style and theme.
2. **Typography Task:** Select appropriate typography for a given theme (e.g., tech startup, children's app).
3. **Layout Design:** Design a simple webpage layout using grid principles to establish hierarchy.

Wireframing and Prototyping

Theory Assignments:

1. **Wireframing Basics:** Explain the importance of wireframes in the design process and differences between low- and high-fidelity wireframes.
2. **Prototype Fundamentals:** Write about the stages of prototyping and its benefits in product development.

Practical Assignments:

1. **Low-Fidelity Wireframe:** Create wireframes for a simple e-commerce page using pencil and paper or a tool like Figma.
2. **Prototype Creation:** Develop a clickable low-fidelity prototype for a mobile app, defining the basic user flows.
3. **Feedback Loop:** Present the prototype to peers and gather feedback, then revise based on suggestions.

Interaction Design and Micro Interaction

Theory Assignments:

1. **Introduction to Interaction Design:** Describe interaction design principles with examples of good and bad interactions.
2. **Microinteractions:** Write about the importance of microinteractions and how they enhance user experience.

Practical Assignments:

1. **Button Animation:** Create animated button states for hover, click, and loading.
 2. **Feedback Interaction:** Design a small feedback interaction, like a form submission animation or an error message.
 3. **Tooltips and Notifications:** Implement tooltips and notification alerts on a mock interface.
-

Ui Design Tools and Practical Application

Theory Assignments:

1. **Tool Comparison:** Compare Figma, Sketch, and Adobe XD in terms of features and ease of use for UI/UX design.
2. **Component-Based Design:** Explain the importance of reusable components and design systems.

Practical Assignments:

1. **Redesign Task:** Redesign a basic webpage or app screen in Figma or Adobe XD.
 2. **Create a Design System:** Build a design system with components like buttons, input fields, and icons.
 3. **Component Library:** Design a library of basic UI elements (buttons, icons, headers) for reuse across projects.
-

Advanced Prototyping and Usability Testing

Theory Assignments:

1. **Usability Testing:** Write a guide on planning and conducting usability testing sessions.
2. **Usability Metrics:** Explain key metrics used to evaluate usability (e.g., task success rate, error rate).

Practical Assignments:

1. **High-Fidelity Prototype:** Create an interactive, high-fidelity prototype for a complex page, like a product checkout flow.
 2. **Conduct a Usability Test:** Test your prototype with a small group, collect feedback, and document findings.
 3. **Refinement:** Refine your design based on the usability test results and document the changes made.
-

Accessibility and Inclusive Design

Theory Assignments:

1. **Accessibility Basics:** Discuss WCAG guidelines and their importance in inclusive design.
2. **Inclusive Design:** Write about how accessibility impacts user experience and the benefits of inclusive design.

Practical Assignments:

1. **Accessibility Audit:** Perform an accessibility audit on a webpage, noting areas for improvement.
2. **Accessible Redesign:** Redesign a form or landing page to improve accessibility, focusing on color contrast, alt text, and keyboard navigation.
3. **Simulated Experience:** Use accessibility tools (e.g., screen reader) to experience a site from the perspective of a visually impaired user.

Mobile and Responsive Design

Theory Assignments:

1. **Responsive vs. Adaptive:** Write a report on the differences between responsive and adaptive design with examples.
2. **Mobile-First Design:** Explain why mobile-first design is critical in modern UX.

Practical Assignments:

1. **Responsive Layout:** Design a simple responsive webpage that adapts to desktop, tablet, and mobile views.
2. **Mobile App Layout:** Create a mobile app layout, focusing on touch-friendly interactions.
3. **Breakpoints Implementation:** Adjust an existing design layout for various screen sizes using breakpoints.

UX Writing and Content Strategy

Theory Assignments:

1. **Importance of UX Writing:** Write about the impact of good microcopy on user experience.
2. **Content Strategy Basics:** Explain the role of content strategy in UI/UX design.

Practical Assignments:

1. **Microcopy Creation:** Write effective microcopy for form fields, error messages, and navigation links.

-
2. **Content Audit:** Conduct a content audit on a small website or app, noting where copy could be improved.
 3. **Develop Content Strategy:** Draft a short content strategy for a hypothetical app or site.
-

Design Handoff and Collaboration

Theory Assignments:

1. **Design Handoff Process:** Explain the handoff process and best practices for working with developers.
2. **Collaboration Tools:** Write about tools (e.g., Zeplin, Figma Inspect) that facilitate collaboration and handoff.

Practical Assignments:

1. **Prepare Design Specs:** Create a design spec document for developers with measurements, colors, and assets.
 2. **Component Documentation:** Document a design component library for developer use.
 3. **Collaborative Project:** Simulate a collaboration where you hand off a design to a “developer” for implementation.
-

Portfolio Development and Career Preparation

Theory Assignments:

1. **Portfolio Best Practices:** Write about what makes a great UI/UX portfolio.
2. **Personal Branding:** Discuss strategies for establishing a unique voice and style as a designer.

Practical Assignments:

1. **Portfolio Creation:** Build a portfolio showcasing your best projects, detailing your design process.
 2. **Project Presentation:** Practice presenting one of your portfolio projects, explaining your design decisions.
 3. **Mock Interview:** Prepare for common interview questions and present a design case study.
-

1. **Complete Product Design:** Choose a real-world problem and design a solution from start to finish, including user research, wireframing, prototyping, usability testing, and final high-fidelity designs.
2. **Document Design Process:** Write a comprehensive case study covering each stage of your design process.
3. **Final Presentation:** Present your project as if pitching to a potential employer or client, receiving feedback on your work and design choices.

Website Designing Assignments

Introduction to Website Design and HTML Basics

Theory Assignments:

1. Explain the structure of an HTML document and the purpose of DOCTYPE.
2. List and explain at least five common HTML tags used in website design.
3. Describe the difference between block-level and inline elements in HTML.

Assignment 1: Basic HTML Document Structure

- **Objective:** Understand the basic structure of an HTML document and practice using common tags.

Instructions:

1. Create a new HTML file.
2. Add the following basic structure:
 - o `<!DOCTYPE html>` declaration.
 - o HTML `<html>`, `<head>`, and `<body>` tags.
3. Inside the `<head>` section, add a title for your webpage.
4. Inside the `<body>` section:
 - o Add a heading (using `<h1>`).
 - o Add a paragraph (using `<p>`).
 - o Add a horizontal line (`<hr>`).
 - o Add another heading (using `<h2>`).
 - o Add a second paragraph.

Assignment 2: Creating Lists and Links

- **Objective:** Practice using list tags and linking to other pages.

Instructions:

1. Inside a new HTML file:
 - o Add a heading with the text "My Favorite Hobbies."
2. Create an ordered list (``) listing three hobbies.
3. Below the ordered list, create an unordered list (``) with three items related to one of the hobbies.
4. Add a link (`<a>`) that directs to a webpage relevant to one of your hobbies.

Assignment 3: Adding Images and Embedding Videos

- **Objective:** Learn how to add images and embed videos in HTML.

Instructions:

1. Create an HTML page with the title “My Travel Memories.”
 2. Inside the `<body>`, add:
 - o A heading (using `<h1>`) that says “My Favorite Travel Destination.”
 - o An image of a travel destination (use the `` tag with an appropriate `src` and `alt`).
 3. Below the image, add a short paragraph describing the place.
 4. Embed a YouTube video of the destination using the `<iframe>` tag.
-

Assignment 4: Building a Simple HTML Form

- **Objective:** Learn how to create forms using HTML.

Instructions:

1. Create an HTML page with the title “Contact Me.”
 2. Inside the `<body>`:
 - o Add a form with the following input fields:
 - Name (text input)
 - Email (email input)
 - Message (textarea)
 - A submit button
 3. Make sure each field has a label using the `<label>` tag for accessibility.
-

Assignment 5: Creating Tables

- **Objective:** Practice using table tags to organize data.

Instructions:

1. Create a new HTML page with a title “Course Schedule.”
 2. Add a table with the following structure:
 - o A header row with three columns: “Day,” “Course Name,” “Time.”
 - o Add three more rows to represent a course schedule with information for each column.
 3. Style the table to have a border and add padding to each cell.
-

Assignment 6: Creating a Personal Profile Page

- **Objective:** Combine different HTML tags to create a basic profile page.

Instructions:

1. Create an HTML file titled “Profile.”
2. Add the following sections:

- **Header:** A heading with your name.
 - **Introduction:** A paragraph about yourself.
 - **Skills:** An unordered list of your skills.
 - **Hobbies:** An ordered list of your hobbies.
 - **Contact:** A link to your email address (using `mailto:`) and a phone number.
3. Add a profile picture using the `` tag with an appropriate `alt` text.
-

Assignment 7: Creating a Navigation Bar

- **Objective:** Practice building a basic navigation bar.

Instructions:

1. Create an HTML page with a title “My Portfolio.”
 2. Inside the `<body>`, create a navigation bar at the top with the following links:
 - Home
 - About Me
 - Portfolio
 - Contact
 3. Use an unordered list (``) to create the navigation bar, and style it with inline CSS to display the links horizontally.
-

Assignment 8: Building a Multi-Page Website

- **Objective:** Practice linking between pages and organizing content across multiple pages.

Instructions:

1. Create a simple 3-page website with the following structure:
 - **Home:** Welcome message and a brief introduction about the website.
 - **About Me:** A paragraph about yourself and an image.
 - **Portfolio:** List of your projects or hobbies with descriptions.
 2. Ensure each page has a link to the other two pages using `<a>` tags.
 3. Include a heading on each page to identify it (e.g., “Welcome to My Homepage”).
-

Assignment 9: Embedding Media and Maps

- **Objective:** Practice embedding media and external content.

Instructions:

1. Create an HTML page with the title “Explore New York City.”
2. Add a paragraph with a short description of New York City.
3. Embed a Google Maps view of New York City using the `<iframe>` tag.
4. Embed a YouTube video about New York City.

Assignment 10: Using Semantic HTML Elements

- **Objective:** Learn and practice using semantic HTML elements.

Instructions:

1. Create an HTML file with the title “Tech News.”
2. Use the following HTML5 semantic tags to structure the page:
 - **Header:** Title of the website.
 - **Nav:** Links to “Home,” “Latest News,” and “Contact.”
 - **Main:** A section with a few news articles.
 - Each article should use the `<article>` tag, with headings and short descriptions.
 - **Footer:** Add contact information in the footer.

Assignment 11: Basic HTML Styling with Inline CSS

- **Objective:** Practice adding inline CSS to style HTML elements.

Instructions:

1. Create an HTML page called “Styled Page.”
2. Use inline CSS to:
 - Change the background color of the page.
 - Center-align the main heading and set a custom font size.
 - Add padding and a border to an image.
3. Add a paragraph with a custom font color and italicized text.

Introduction to CSS and Styling Basics

Theory Assignments:

1. Explain the difference between inline, internal, and external CSS.
2. Describe CSS selectors and list the types of selectors (e.g., element, class, id).
3. Discuss the CSS box model and its components.

Assignment 1: Basic Text Styling

- **Objective:** Practice basic text styling using CSS.

Instructions:

1. Create an HTML page with a title and a paragraph.
2. In a separate CSS file or within `<style>` tags:
 - Set the font size of the title to 24px.
 - Change the font family of the paragraph text to Arial.
 - Set the color of the text to navy.
 - Center-align the title and justify the paragraph text.

Assignment 2: Adding Backgrounds and Colors

- **Objective:** Practice using colors and background properties.

Instructions:

1. Create an HTML page with three sections (`<div>` elements) labeled “Introduction,” “Content,” and “Footer.”
2. Apply CSS to:
 - o Give each section a different background color (use HEX or RGB values).
 - o Add padding and set a specific width for each section.
 - o Use a gradient background for the “Introduction” section.
 - o Set a light gray background color for the entire webpage.

Assignment 3: Working with Borders and Shadows

- **Objective:** Learn how to use borders, rounded corners, and shadows.

Instructions:

1. Create a simple HTML page with a card-like structure.
2. Add CSS to style the card:
 - o Add a border with a color, style, and width of your choice.
 - o Round the corners using `border-radius`.
 - o Apply a box shadow to give it a raised look.

Assignment 4: Creating a Simple Navigation Bar

- **Objective:** Style a basic navigation bar using CSS.

Instructions:

1. Create a navigation bar with links to “Home,” “About,” “Services,” and “Contact.”
2. In your CSS:
 - o Make the links horizontal by setting `display: inline-block`.
 - o Add padding and a background color to each link.
 - o Remove the underline and set a hover effect to change the link color.
 - o Center-align the entire navigation bar on the page.

Assignment 5: Using Flexbox for Layout

- **Objective:** Learn to use Flexbox for creating responsive layouts.

Instructions:

1. Create an HTML file with a header, main content area, and a footer.
2. Style the layout using Flexbox:
 - o Set up the main container as a Flexbox container.
 - o Arrange the elements in a column layout.
 - o Center-align the content horizontally.
 - o Adjust the layout to a row for screens wider than 768px.

Assignment 6: Grid Layout

- **Objective:** Use CSS Grid for creating a gallery layout.

Instructions:

1. Create an HTML page with a gallery of 6 images.
2. Apply CSS Grid to:
 - o Arrange the images in a 3x2 grid layout.
 - o Set a gap between the grid items.
 - o Adjust the grid to be 2x3 for screens narrower than 600px.
 - o Center the entire gallery on the page.

Assignment 7: Styling Buttons

- **Objective:** Practice styling buttons with CSS.

Instructions:

1. Create three buttons in HTML: "Primary," "Secondary," and "Danger."
2. Style each button in CSS:
 - o Set distinct background colors for each button.
 - o Add padding, rounded corners, and a shadow effect.
 - o Change the button color on hover.
 - o Add transitions to make the hover effect smooth.

Assignment 8: Responsive Design with Media Queries

- **Objective:** Learn how to make a responsive layout using media queries.

Instructions:

1. Create an HTML page with a main container containing a header, three content sections, and a footer.
2. Apply CSS to:
 - o Make the layout a single column on screens narrower than 768px.
 - o Change the layout to two columns for screens wider than 768px.
 - o Adjust font sizes and padding at different screen widths.

Assignment 9: Styling a Contact Form

- **Objective:** Practice styling forms with CSS.

Instructions:

1. Create a simple contact form with fields for name, email, and message.
2. In your CSS file:
 - o Style each form field with padding, border, and background color.
 - o Style the submit button with a distinct color, padding, and rounded corners.
 - o Add focus effects for input fields.
 - o Center the form on the page

Assignment 10: Using Pseudo-Classes and Pseudo-Elements

- **Objective:** Learn how to use pseudo-classes and pseudo-elements for styling.

Instructions:

1. Create an HTML page with several links and a paragraph.
2. Apply CSS to:
 - Style links with different colors for hover, active, and visited states.
 - Use `::first-line` to style only the first line of the paragraph.
 - Add a decorative `::before` element to headings.

Assignment 11: Using CSS Variables

- **Objective:** Practice using CSS variables to maintain color and theme consistency.

Instructions:

1. Create a simple web page with a header, main content, and footer.
2. Define CSS variables for primary color, secondary color, and text color.
3. Apply the variables to style the background, text, and border colors.
4. Adjust the variables to see how the page's color theme changes.

Assignment 12: Creating Animations

- **Objective:** Learn how to create basic CSS animations.

Instructions:

1. Create an HTML page with a box (using a `<div>`).
2. Create an animation in CSS:
 - Make the box move from left to right.
 - Add a color change as the box moves.
 - Set the animation to loop indefinitely.

Assignment 13: Building a Product Card

- **Objective:** Combine various CSS skills to style a product card.

Instructions:

1. Create a product card in HTML with an image, title, description, and a price.
2. In your CSS:
 - Set the layout using Flexbox or Grid.
 - Style the title, price, and description.
 - Add a hover effect to scale up the card slightly.
 - Add a shadow to the card.

Assignment 14: Styling a Landing Page with Custom Fonts and Icons

- **Objective:** Use custom fonts and icons to enhance a page's look.

Instructions:

1. Create an HTML landing page with sections for "Hero," "Features," and "Contact Us."
 2. In your CSS:
 - Use Google Fonts to import and apply a custom font to headings.
 - Use Font Awesome (or a similar library) to add icons to the "Features" section.
 - Style each section with padding, background colors, and font adjustments.
-

Assignment 15: Styling a Landing Page with Custom Fonts and Icons

- **Objective:** Practice using CSS transformations and transitions for interactive effects.

Instructions:

1. Create an HTML page with an image gallery.
2. In CSS:
 - Apply a transition effect to each image that changes its scale and adds a shadow when hovered.
 - Use `transform` properties to rotate or scale the images.
 - Ensure the effects are smooth by adding a duration to the transition.

Practical Assignments

1. Style the HTML webpage created in Topic 1 by:
 - Adding colors, font styles, and padding/margin to each section.
 - Experimenting with background colors and borders.
2. Create a navigation menu with horizontal and vertical layouts using CSS.

Assignment 16: Responsive Design with Media Queries

Theory Assignment:

1. Define responsive design and its importance in modern web development.
2. Explain the role of media queries in responsive design.
3. Describe how `viewport` settings affect mobile displays.

Practical Assignment:

1. Make your HTML page from Topic 1 responsive by:
 - Using media queries to adjust the layout for mobile screens.
 - Hiding/showing elements or adjusting font sizes for smaller screens.
2. Create a simple layout with a sidebar that moves below the main content on screens smaller than 768px.

Assignment 17: Introduction to Bootstrap

Theory Assignment:

1. What is Bootstrap, and why is it useful for website design?
2. Explain the Bootstrap grid system and how it helps create responsive layouts.
3. List and explain at least three Bootstrap components (e.g., navbar, cards, buttons).

Practical Assignment:

1. Redesign the HTML webpage using Bootstrap to:
 - Implement a responsive grid layout for the header, main content, and footer.
 - Add Bootstrap buttons and a styled form using Bootstrap classes.
 2. Create a simple portfolio page with Bootstrap's card component to display portfolio items in a grid format.
-

Assignment 18: Advanced Bootstrap Components

Theory Assignment:

1. Explain how modals and carousels work in Bootstrap.
2. Describe the purpose of utility classes in Bootstrap and give examples.
3. Discuss the importance of customizing Bootstrap variables for unique styling.

Practical Assignment:

1. Add a Bootstrap carousel to showcase multiple images on your portfolio page.
 2. Create a contact form in a modal that opens on a button click.
 3. Customize Bootstrap using variables (e.g., changing primary colors and button styles).
-

Assignment 19: Introduction to Tailwind CSS

Theory Assignment:

1. Explain what Tailwind CSS is and how it differs from traditional CSS frameworks.
2. Describe the concept of utility-first CSS and its advantages.
3. List and explain at least five common Tailwind classes.

Practical Assignment:

1. Redesign your HTML webpage using Tailwind CSS by:
 - Applying utility classes for styling and layout adjustments.
 - Ensuring it is responsive with Tailwind's responsive utilities.
 2. Create a pricing table using Tailwind with three columns for different pricing options, including buttons and card elements.
-

Assignment 20: Advanced Tailwind CSS Components

Theory Assignment:

1. Explain how Tailwind's configuration file works and its role in customizing Tailwind.
2. Describe how to create responsive designs using Tailwind's breakpoints.
3. Discuss using custom colors and spacing with Tailwind's configuration.

Practical Assignment:

1. Customize Tailwind's configuration to include a new color scheme and spacing values.
2. Create a custom button and card component using Tailwind and your custom configurations.
3. Design a login page layout with Tailwind, including a form and a styled submit button.

Assignment 21: Advanced CSS: Flexbox

Theory Assignment:

1. Explain the purpose of Flexbox and its benefits for responsive design.
2. Describe the main properties of Flexbox (`flex-direction`, `justify-content`, `align-items`).
3. Discuss the difference between `flex-grow`, `flex-shrink`, and `flex-basis`.

Practical Assignment:

1. Create a responsive navigation bar with Flexbox that aligns items horizontally and adjusts to a vertical layout on mobile screens.
 2. Design a three-column layout using Flexbox with each column equally spaced.
 3. Build a product card layout with Flexbox where elements (image, title, description, and price) align and space evenly.
-

Assignment 22: Advanced CSS: CSS Grid

Theory Assignment:

1. What is CSS Grid, and how is it different from Flexbox?
2. Describe the CSS Grid properties `grid-template-columns` and `grid-template-rows`.
3. Explain the purpose of `grid-area` and how it is used to create complex layouts.

Practical Assignment:

1. Create a responsive grid layout for an image gallery with CSS Grid, displaying images in a 3-column layout on desktop and a 1-column layout on mobile.
 2. Design a multi-section page layout using CSS Grid, where sections like header, sidebar, main content, and footer are arranged in a grid.
 3. Build a blog post layout using Grid, with a main content area, related posts sidebar, and footer section.
-

Assignment 23: Advanced CSS: Sass and Less

Theory Assignment:

1. Describe the benefits of using a CSS preprocessor like Sass or Less.
2. Explain variables, nesting, and mixins in Sass.
3. Describe the purpose of inheritance and partials in Sass.

Practical Assignment:

1. Convert the CSS styling of one of your previous assignments (e.g., portfolio page) to Sass by:
 - o Using variables for colors, font sizes, and spacing.
 - o Applying nesting and mixins to simplify the CSS structure.
2. Create a Sass file with partials for typography, layout, and colors, then import them into a main stylesheet.
3. Write a mixin in Sass for a responsive card component that can adjust its layout based on screen size.

WordPress Assignments

1. Introduction to WordPress

Theory Questions:

1. Explain what WordPress is and list some of its key features.
2. What is the difference between WordPress.com and WordPress.org?
3. Why is WordPress a popular choice for building websites?
4. List five types of websites that can be created using WordPress.

Practical Assignment:

1. **Installation:** Set up a WordPress site on your local machine using XAMPP or a similar server environment.
2. **Basic Site Setup:** Once WordPress is installed, set up a new site, add a title, tagline, and logo for the site, and create a homepage.

2. WordPress Dashboard and Settings

Theory Questions:

1. Describe the purpose of the WordPress dashboard. What are some common sections?
2. Explain the importance of permalink settings and how they affect SEO.
3. What are themes and plugins, and how are they different?

Practical Assignment:

1. **Site Settings:** Explore and configure the basic settings in the WordPress dashboard, including General, Reading, and Permalinks.
2. **Theme Installation:** Install a free theme from the WordPress theme repository and activate it on your site.

3. Posts, Pages, and Categories

Theory Questions:

1. What is the difference between a post and a page in WordPress?
2. Explain the purpose of categories and tags for organizing content.
3. Describe the main post formats available in WordPress.

Practical Assignment:

- **Exercise 1:** Create three different blog posts on topics of your choice. Use different post formats (Standard, Image, Video) for each post. For each post, add a title, featured image, and content.
- **Exercise 2:** For one of the posts, experiment with formatting options:

- Add a list of bullet points.
 - Use heading tags (H2, H3) within the post.
 - Apply text styling (bold, italics, underline).
 - Add a link to an external website within the content.
 - **Exercise 3:** Create a new post that includes the following elements:
 - An embedded YouTube video.
 - An image gallery of at least three images.
 - Ablockquote.
 - A call-to-action button linking to a contact page.
-

2. Organizing Content with Categories and Tags

- **Exercise 1:** Create five new blog posts on different topics related to a theme (e.g., travel, food, technology). Use unique tags for each post to help organize content.
 - **Exercise 2:** Create three categories (e.g., "News," "Tutorials," "Events"). Assign each blog post from the previous exercise to a relevant category and add suitable tags.
 - **Exercise 3:** Add a category and tag description to each category and tag created. Explain how you can display these descriptions on your site.
-

3. Managing Pages and Content Structure

- **Exercise 1:** Create three static pages:
 - An "About Us" page that includes a brief description of your site.
 - A "Services" page with a list of at least three services you offer.
 - A "Contact Us" page with a contact form (use Contact Form 7 or a similar plugin).
 - **Exercise 2:** Set one of your pages (e.g., "About Us") as the homepage of your WordPress site and another (e.g., "Blog") as the posts page in the settings.
 - **Exercise 3:** Create a parent page called "Portfolio" and two child pages under it, named "Web Design Projects" and "Graphic Design Projects." Add sample content to each page to represent different projects.
-

4. Advanced Post and Page Customization

- **Exercise 1:** Use the WordPress Block Editor (Gutenberg) to create a visually appealing post layout:
 - Use columns to display content side by side.
 - Add a cover image with text overlay as the post header.
 - Experiment with color settings for different blocks to enhance the visual appeal.
- **Exercise 2:** Create a "Resources" page that lists downloadable files or links. Include at least five downloadable items (e.g., PDF files or links to resources), and add descriptions for each item.
- **Exercise 3:** Create a landing page using the page editor. This page should include:
 - A full-width cover image.
 - A call-to-action button.

- An email subscription form (use a plugin like Mailchimp).
-

5. Customizing Post Visibility and Publication Settings

- **Exercise 1:** Create a blog post and set its visibility to “Private.” Then log out and try to view the post as a non-logged-in visitor to understand visibility restrictions.
 - **Exercise 2:** Create a post and schedule it to publish on a specific date and time in the future. Verify the post publishes automatically on the scheduled date.
 - **Exercise 3:** Write two versions of a blog post and use the “Pending Review” option for one version. Share this version with another user (if available) for review.
-

6. Working with Featured Images and Excerpts

- **Exercise 1:** Create a new post and upload a featured image. Experiment with different image sizes and observe how the theme displays these featured images in the blog feed and single post view.
 - **Exercise 2:** Add a custom excerpt to a post. Ensure the excerpt gives a concise summary of the content. Check how the excerpt displays on the main blog page.
 - **Exercise 3:** Write a post with a long introduction, then use the "Read More" tag to break up the content. Verify how the post appears in the blog feed with the "Read More" link.
-

7. Creating Custom Post Categories and Tag Clouds

- **Exercise 1:** Add five new categories that are specific to different types of content (e.g., “Guides,” “Tips,” “News,” “Reviews,” “Interviews”).
 - **Exercise 2:** Display a category list in the sidebar using the “Categories” widget. Configure it to show the post count for each category.
 - **Exercise 3:** Add a tag cloud widget to the sidebar to display the most popular tags. Ensure each post has at least two unique tags to populate the tag cloud effectively.
-

8. Content Organization and Cleanup

- **Exercise 1:** Organize all of your posts into relevant categories and tags. Ensure there are no uncategorized posts.
 - **Exercise 2:** Merge two similar categories into one (e.g., merge “Guides” and “Tutorials” into “How-To Guides”). Update posts with the new category.
 - **Exercise 3:** Use the “Quick Edit” feature to change the author, categories, and tags for multiple posts. Document any improvements you make to your post organization.
-

4. Themes and Customization

Theory Questions:

1. What is a WordPress theme, and how does it impact the look and functionality of a site?
2. Explain what the WordPress Customizer is used for.
3. What are child themes, and why might you use one?

Practical Assignment:

- **Exercise 1:** Go to the WordPress theme repository and search for a theme that suits a personal blog. Install and activate this theme. Take note of the key features that make it suitable for a blog.
 - **Exercise 2:** Install a second theme that is designed for a business website. Activate it and compare its structure with the blog theme you installed earlier. List any significant layout or feature differences.
 - **Exercise 3:** Using a local WordPress installation, install and activate three different themes. Explore each one and describe how the main menu, homepage layout, and footer differ between them.
-

2. Basic Theme Customization Using the WordPress Customizer

- **Exercise 1:** Open the WordPress Customizer and change the site title, tagline, and logo of your website. Observe how these changes reflect in real time.
 - **Exercise 2:** Customize the header by changing the header background color or image. Experiment with different header layouts if your theme supports it.
 - **Exercise 3:** Customize the homepage by modifying colors for the background, text, and links. Apply changes that make the theme more personalized.
-

3. Adding and Configuring Widgets

- **Exercise 1:** Add the “Recent Posts” widget to your sidebar and configure it to display the five latest posts.
 - **Exercise 2:** Add a “Categories” widget to the footer. Configure it to display post counts and a dropdown list of categories.
 - **Exercise 3:** Use the “Text” widget to add a short welcome message in the sidebar of your website. Use some basic HTML tags to format the text (e.g., bold or italics).
-

4. Creating and Customizing Menus

- **Exercise 1:** Create a primary menu with links to Home, About, Blog, and Contact Us pages. Set this menu as the primary navigation menu.
- **Exercise 2:** Create a footer menu that includes links to Privacy Policy and Terms of Service. Set it as the footer menu in your theme.

- **Exercise 3:** Create a nested (dropdown) menu with parent and child items. For example, make "Services" a parent item with child items such as "Web Design" and "SEO Services." Experiment with how the dropdown appears in your theme.
-

5. Using Custom Page Templates

- **Exercise 1:** Using a theme with multiple page templates, create an "About Us" page. Select a custom page template that has a different layout (e.g., full-width or sidebar layout) from the default.
 - **Exercise 2:** Create a "Gallery" page and set it to use a page template with no sidebar (if available). Add some images to this page to test the layout.
 - **Exercise 3:** Create a "Contact Us" page and assign a template specifically designed for contact information. Use the Customizer to add this page to your main menu.
-

6. Customizing the Homepage Layout

- **Exercise 1:** Set up a static homepage by creating a new page called "Home." Assign this page as your homepage in the settings and add a few paragraphs of content to test the layout.
 - **Exercise 2:** Set up a blog page by creating a page called "Blog." Assign it as your posts page in the settings and make sure it displays recent blog posts.
 - **Exercise 3:** Add a featured image to your homepage and test how it displays on your theme. Adjust the image size or aspect ratio to see how it affects the homepage appearance.
-

7. Using Custom CSS for Theme Styling

- **Exercise 1:** Use the Additional CSS section in the Customizer to change the background color of the website header. Document the CSS code used.
 - **Exercise 2:** Change the font color and size for all h1 headings on the site using custom CSS. Test the changes on multiple pages to verify consistency.
 - **Exercise 3:** Use CSS to add a border around all images on the site. Test the effect by adding images to different pages and posts.
-

8. Exploring Theme Options and Settings

- **Exercise 1:** Explore the theme options under Appearance > Customize. Document any unique options such as layout choices, color schemes, or typography settings.
- **Exercise 2:** Change the typography settings for your site (e.g., change the font family and size of headings and body text). Observe how different fonts affect the look and readability of your site.
- **Exercise 3:** Customize the layout settings to make your website's homepage display in a grid layout instead of a list layout (if supported by your theme).

9. Installing and Using Child Themes

- **Exercise 1:** Create a child theme of your current theme. Activate the child theme and verify that it inherits the styles and functions of the parent theme.
 - **Exercise 2:** Modify the style.css file of the child theme to change the color of links across the site. Document your changes and verify that they apply site-wide.
 - **Exercise 3:** Add a new functions.php file to the child theme and write a simple function to add custom text in the footer (e.g., “Powered by [Your Name]”). Verify that the text appears in the footer.
-

10. Exploring and Customizing Theme Header and Footer

- **Exercise 1:** Use the WordPress Customizer to add a header image and adjust its alignment, size, or position. Check how this affects the site’s appearance on different devices.
 - **Exercise 2:** Customize the footer by adding or removing widgets (if supported by your theme). Add a “Text” widget with a short copyright notice.
 - **Exercise 3:** Use the Additional CSS option to change the footer background color and text color to make it match your site’s branding.
-

11. Making the Theme Responsive

- **Exercise 1:** Using the Customizer preview, view your website on different device sizes (desktop, tablet, mobile). Take notes on any design elements that don’t adjust well.
 - **Exercise 2:** If your theme allows it, customize the mobile menu layout and test the appearance on a mobile device or emulator.
 - **Exercise 3:** Use custom CSS to adjust the font size of paragraph text on smaller screens (e.g., tablets and mobile). Document the CSS code and verify it works on different screen sizes.
-

12. Exploring and Using Page Builders (Bonus Exercise)

- **Exercise 1:** Install and activate a page builder plugin (e.g., Elementor or WPBakery). Create a new page and use the page builder to design a unique layout with sections, columns, and elements.
 - **Exercise 2:** Using the page builder, create a “Services” page with image icons for each service, accompanied by descriptions and call-to-action buttons.
 - **Exercise 3:** Explore the responsive editing options in the page builder to adjust your page layout for mobile and tablet views. Preview how the layout adapts to different screen sizes.
-

5. Plugins and Functionality Extensions

Theory Questions:

1. What is a plugin in WordPress, and what are the benefits of using them?
2. List three essential plugins for WordPress websites and explain their use.
3. What is the difference between free and premium plugins?

Practical Assignment:

- **Exercise 1:** Install and activate a popular SEO plugin (e.g., Yoast SEO or All in One SEO). Go through the basic setup wizard and configure it to optimize your site for search engines.
 - **Exercise 2:** Install and activate a contact form plugin (e.g., Contact Form 7 or WPForms). Create a simple contact form and add it to a Contact Us page on your website.
 - **Exercise 3:** Install a caching plugin (e.g., W3 Total Cache or WP Super Cache) and configure it to improve the performance of your website. Clear the cache and verify that changes are reflected on the website.
-

2. Customizing Website Security with Plugins

- **Exercise 1:** Install and activate a security plugin (e.g., Wordfence Security or iThemes Security). Perform a basic security scan and identify any vulnerabilities.
 - **Exercise 2:** Configure the plugin to enable two-factor authentication (2FA) for the login page if available. Test the 2FA feature by logging in with a secondary verification step.
 - **Exercise 3:** Set up basic firewall protection using the security plugin and configure login attempt limits. Try to log in multiple times with incorrect credentials to see how the plugin responds.
-

3. Extending Functionality with E-commerce Plugins

- **Exercise 1:** Install and activate WooCommerce. Configure the basic settings for a new online store, such as currency, payment options, and shipping.
 - **Exercise 2:** Add two simple products to your WooCommerce store, including images, descriptions, prices, and stock status. Preview the product pages to see how they appear to customers.
 - **Exercise 3:** Install a WooCommerce-related plugin (e.g., WooCommerce PayPal Payments) and set up an additional payment option. Verify that the payment option is visible on the checkout page.
-

4. Social Media Integration Using Plugins

- **Exercise 1:** Install and activate a social sharing plugin (e.g., Social Snap or Sassy Social Share). Configure it to display sharing buttons at the bottom of blog posts.
- **Exercise 2:** Add social media follow buttons to your website's sidebar or footer using the same plugin or a different one, like Smash Balloon Social Photo Feed. Customize the button styles to match your theme.
- **Exercise 3:** Install and set up an Instagram feed plugin to display recent posts from a sample Instagram account on your website's homepage or sidebar.

5. Enhancing Performance with Image Optimization Plugins

- **Exercise 1:** Install and activate an image optimization plugin (e.g., Smush or Imagify). Bulk optimize the images on your website to reduce file size and improve page load times.
 - **Exercise 2:** Upload a few high-resolution images to your media library and use the plugin to compress them. Compare the file size before and after compression.
 - **Exercise 3:** Configure the plugin to automatically compress and resize images upon upload. Upload a new image to test this functionality and verify the compression.
-

6. Creating and Displaying Custom Post Types

- **Exercise 1:** Install and activate the Custom Post Type UI plugin. Create a custom post type called “Portfolio” and configure it to display in the admin menu.
 - **Exercise 2:** Add a new portfolio item using your custom post type, including a title, featured image, and description. Preview how it displays on the site.
 - **Exercise 3:** Use the custom post type to create three additional portfolio items. Display them in a grid layout on a dedicated “Portfolio” page.
-

7. Using Page Builder Plugins for Custom Layouts

- **Exercise 1:** Install and activate a page builder plugin (e.g., Elementor or Beaver Builder). Create a custom homepage layout with sections for an introduction, services, and a contact form.
 - **Exercise 2:** Using the page builder, create a “Team” page that displays team members in columns. Each team member should include a photo, name, and position.
 - **Exercise 3:** Design a landing page for a product or service with the page builder. Include sections for a product description, features, testimonials, and a call-to-action button.
-

8. Managing Website Backups Using Plugins

- **Exercise 1:** Install and activate a backup plugin (e.g., UpdraftPlus or BackWPup). Create a manual backup of your website and store it locally or in cloud storage.
 - **Exercise 2:** Set up an automatic backup schedule to run weekly backups of your site. Verify that the schedule is working by checking your backup history.
 - **Exercise 3:** Download the backup file and simulate a restore process in a local WordPress environment to test the backup functionality.
-

9. Improving Accessibility with Plugins

- **Exercise 1:** Install and activate an accessibility plugin (e.g., WP Accessibility). Enable skip links, screen reader text, and keyboard navigation support.
 - **Exercise 2:** Configure the plugin to add alt text to images and check for any issues in your media library. Modify alt text where necessary for better accessibility.
 - **Exercise 3:** Run an accessibility scan using the plugin and address any issues flagged. For example, fix color contrast issues or missing form labels.
-

10. Adding Custom Fields Using Advanced Custom Fields (ACF)

- **Exercise 1:** Install and activate the Advanced Custom Fields (ACF) plugin. Create a custom field group for “Portfolio” posts that includes fields for “Client Name” and “Project Date.”
 - **Exercise 2:** Add custom field data to a portfolio item and display it on the frontend of your website. Test by creating multiple entries with different field values.
 - **Exercise 3:** Use the ACF plugin to create a custom field group for blog posts, with fields for “Author Bio” and “Reading Time.” Display these fields at the end of each post.
-

11. Adding Analytics with Google Site Kit

- **Exercise 1:** Install and activate the Google Site Kit plugin. Connect it to Google Analytics and verify your site to begin tracking traffic.
 - **Exercise 2:** View the Google Analytics dashboard within WordPress to see your website’s traffic overview. Record key metrics, such as page views and session duration.
 - **Exercise 3:** Set up Google Search Console within Site Kit to view your site’s search performance. Check for search queries and pages that bring traffic to your site.
-

12. Enhancing Email Marketing with Plugins

- **Exercise 1:** Install and activate an email marketing plugin (e.g., Mailchimp for WordPress). Create a signup form and add it to your website’s sidebar or footer.
 - **Exercise 2:** Customize the email signup form’s appearance to match your website’s branding. Test the form to ensure users are correctly added to the mailing list.
 - **Exercise 3:** Integrate a popup plugin (like Popup Maker) with your email signup form. Configure a popup that appears when users scroll down the page or after a set time.
-

13. Translation and Multilingual Capabilities

- **Exercise 1:** Install and activate a translation plugin (e.g., Polylang or WPML). Configure it to add a secondary language to your website.
- **Exercise 2:** Translate a few pages and posts into the new language. Verify that users can switch between languages using the language selector.

- **Exercise 3:** Customize the language switcher's appearance to fit your theme's style. Test the site on both desktop and mobile to ensure the switcher is accessible.
-

14. Creating Custom Login Pages

- **Exercise 1:** Install and activate a custom login plugin (e.g., LoginPress). Customize the login page with your site's logo and a branded color scheme.
 - **Exercise 2:** Add a background image to the login page and test the layout for responsiveness.
 - **Exercise 3:** Use the plugin's settings to add a "Lost Password" link and configure a custom login URL. Verify that the custom login page works by logging out and logging in again.
-

6. Working with Menus and Widgets

Theory Questions:

1. Describe how menus are used in WordPress and why they are important.
2. What are widgets in WordPress, and where can they be displayed?
3. How do you add a custom menu to a sidebar using widgets?

Practical Assignment:

Creating and Customizing Menus

- **Exercise 1:** Create a primary navigation menu with the following items: Home, About Us, Services, Blog, and Contact Us. Set this menu as the main menu for the website and link each item to the corresponding page.
 - **Exercise 2:** Add a custom link to the primary menu that links to an external site (e.g., a social media profile). Open this link in a new tab and style it differently by adding a custom class for unique styling.
 - **Exercise 3:** Organize the Services page as a parent menu item with three child items: "Web Development," "SEO Services," and "Digital Marketing." Make sure the submenu items appear on hover and test that each link navigates to the correct page.
-

2. Creating a Footer Menu

- **Exercise 1:** Create a footer menu with links to Privacy Policy, Terms of Service, and Contact Us. Display this menu in the footer widget area.
 - **Exercise 2:** Customize the footer menu's appearance using CSS to make the links smaller and in a single horizontal line. Test the responsiveness of the footer menu on mobile devices.
 - **Exercise 3:** Add social media links (e.g., Facebook, Twitter, LinkedIn) to the footer menu and display social media icons beside each link.
-

3. Creating a Secondary or Sidebar Menu

- **Exercise 1:** Create a secondary menu for your website's blog section. Include links to recent blog categories, such as "Latest Articles," "Tutorials," and "News."
 - **Exercise 2:** Place the secondary menu in the sidebar widget area of the blog. Style it to make it look distinct from the main navigation menu.
 - **Exercise 3:** Customize the sidebar menu's styling, such as background color, text color, and hover effects, using custom CSS in the theme customizer.
-

4. Customizing Menu Appearance with CSS

- **Exercise 1:** Add custom CSS to change the primary menu's background color on hover. Make sure the hover effect is visually noticeable and consistent across all menu items.
 - **Exercise 2:** Customize the submenu's font size, padding, and alignment to make it distinct from the main menu items. Test the submenu's appearance on different screen sizes.
 - **Exercise 3:** Create a "sticky" menu that remains at the top of the screen when users scroll down. Use CSS to apply a shadow effect to the menu bar when it becomes sticky.
-

5. Creating and Using Mega Menus

- **Exercise 1:** Install a mega menu plugin (such as Max Mega Menu) and configure it to show multiple columns under the Services menu. Populate the columns with subcategories, such as "Web Development" and "Digital Marketing."
 - **Exercise 2:** Add icons to each submenu item within the mega menu. Use icons that visually represent each service (e.g., a computer icon for "Web Development").
 - **Exercise 3:** Customize the background color, font, and hover effects for the mega menu to match the site's theme. Test it for responsiveness across mobile and desktop devices.
-

Practical Assignments for Widgets

1. Adding Basic Widgets to Sidebar and Footer

- **Exercise 1:** Add a "Search" widget to the sidebar and a "Recent Posts" widget to display the latest five blog posts.
 - **Exercise 2:** Add a "Categories" widget to the sidebar with a dropdown style. Enable the option to display post counts beside each category.
 - **Exercise 3:** In the footer widget area, add a "Text" widget with a short description or an "About Us" section for the website. Customize the text style using CSS to match the site's branding.
-

2. Using Custom HTML Widgets

- **Exercise 1:** Add a "Custom HTML" widget to the sidebar to display an HTML code snippet for a banner ad or an image link. Style the banner to fit within the sidebar dimensions.
- **Exercise 2:** Create a "Custom HTML" widget to display a subscription form using basic HTML and CSS. Style the form fields to match the website's color scheme.

- **Exercise 3:** Use the “Custom HTML” widget to display an embedded Google Map of your business location in the footer. Customize the width and height of the map to fit the footer area.
-

3. Creating a Social Media Feed Widget

- **Exercise 1:** Install a social media feed plugin (e.g., Smash Balloon Social Photo Feed) and add a widget displaying an Instagram feed in the sidebar.
 - **Exercise 2:** Customize the feed layout (e.g., grid style, number of posts displayed) and add a “Follow us on Instagram” button below the feed.
 - **Exercise 3:** Add a Twitter feed widget to the footer area to display recent tweets. Customize the text color and background to match the website’s branding.
-

4. Displaying Recent Comments and Popular Posts Widgets

- **Exercise 1:** Add a “Recent Comments” widget to the sidebar and configure it to show the latest five comments with commenters’ names and a short excerpt of each comment.
 - **Exercise 2:** Install and configure a “Popular Posts” plugin to display the most-viewed posts in the sidebar. Set it to show post titles and thumbnails.
 - **Exercise 3:** Customize the widget title and content styling for both widgets to make them visually cohesive with the rest of the sidebar.
-

5. Customizing Widgets with CSS

- **Exercise 1:** Style the “Recent Posts” widget by adding custom CSS to change the font color, size, and padding of each post title.
 - **Exercise 2:** Add a custom background color and border radius to each widget in the sidebar. Apply a shadow effect on hover.
 - **Exercise 3:** Customize the “Categories” widget to display as a horizontal list with spacing between each category. Test the styling for responsiveness on mobile devices.
-

6. Creating a Custom Widget Area

- **Exercise 1:** Register a custom widget area (e.g., for the header) in your theme by modifying the theme’s functions.php file. Ensure this new widget area is available in the WordPress admin panel.
 - **Exercise 2:** Add a “Search” widget and a “Social Media” widget to this custom widget area. Display it at the top of your homepage.
 - **Exercise 3:** Style the custom widget area with CSS to blend it seamlessly into the header design. Test the layout on both mobile and desktop views.
-

7. Using Image and Gallery Widgets

- **Exercise 1:** Add an “Image” widget to the sidebar and upload a promotional banner. Link the image to a relevant page, such as a sales or promotions page.

- **Exercise 2:** Add a “Gallery” widget to the footer to display a set of images showcasing recent projects or events. Configure it to display as a grid.
 - **Exercise 3:** Style the images in the gallery to have rounded corners and a hover effect. Test the gallery’s appearance on various screen sizes.
-

8. Adding and Customizing RSS Feed Widgets

- **Exercise 1:** Add an “RSS” widget to the sidebar to display a feed from an external blog or news site related to the website’s niche.
 - **Exercise 2:** Set the RSS widget to display the title and excerpt of each post. Limit the feed to show the latest three items.
 - **Exercise 3:** Customize the RSS widget styling, including font size, color, and spacing, to match the design of the website.
-

9. Conditional Widgets (Displaying Widgets on Specific Pages)

- **Exercise 1:** Use a plugin like Widget Options to display a “Related Posts” widget only on single blog post pages.
 - **Exercise 2:** Configure the “Contact Info” widget to display only on the Contact Us page using the plugin settings.
 - **Exercise 3:** Test the conditional visibility by navigating to various pages and verifying that the widgets appear only where configured.
-

7. Custom Post Types and Custom Fields (Basic)

Theory Questions:

1. What is a custom post type in WordPress, and how is it different from standard posts and pages?
2. What are custom fields, and how are they useful in WordPress?

Practical Assignment:

1. **Custom Post Type Creation:** Use a plugin (e.g., CPT UI) to create a new custom post type called “Portfolio.” Add sample content to demonstrate its usage.
 2. **Custom Fields:** Use the Advanced Custom Fields (ACF) plugin to add custom fields to your Portfolio post type. For example, add a field for project date, client name, and project URL.
-

8. Responsive Design with Bootstrap or Tailwind

Theory Questions:

1. Why is responsive design important for WordPress websites?
2. Explain the difference between Bootstrap and Tailwind CSS frameworks.
3. Describe the process of integrating Bootstrap or Tailwind into a WordPress theme.

Practical Assignment:

1. **Responsive Header:** Customize your WordPress theme's header to be responsive using either Bootstrap or Tailwind classes.
 2. **Custom CSS:** Create a responsive two-column layout on the homepage using either Bootstrap or Tailwind.
-

9. WordPress Security and Backup

Theory Questions:

1. List three common security practices to protect a WordPress site.
2. Explain the purpose of backing up a WordPress site.
3. What are the roles of SSL certificates and HTTPS in WordPress security?

Practical Assignment:

1. **Install Security Plugin:** Install and configure a basic security plugin (e.g., Wordfence or iThemes Security) and activate recommended security settings.
 2. **Backup Plugin:** Install a backup plugin (e.g., UpdraftPlus), create a backup of the site, and download it locally.
-

10. SEO Basics for WordPress

Theory Questions:

1. What is SEO, and why is it important for WordPress websites?
2. Describe the purpose of meta tags and keywords in SEO.
3. What role do sitemaps and permalinks play in SEO?

Practical Assignment:

1. **Optimize Content:** Use the Yoast SEO plugin to optimize one of your posts for a target keyword. Ensure the SEO and readability indicators turn green.
 2. **Create a Sitemap:** Generate a sitemap for your WordPress site and submit it to Google Search Console (explain the process).
-

11. Final Project

Project Requirements:

1. **Create a Small Business Website:** Design a small business website (e.g., for a cafe or photography portfolio) with the following pages:
 - o Home
 - o About
 - o Services or Portfolio
 - o Blog

- Contact
- 2. **Use Customization Features:** Customize the theme, add a custom menu, and apply a consistent color scheme.
- 3. **Responsive Design:** Ensure the website is responsive and looks good on both desktop and mobile devices.
- 4. **SEO and Security:** Optimize the site for SEO and set up basic security measures.
- 5. **Submission Requirements:** Submit screenshots of the site along with a short document describing the customization choices and plugins used.