**Week 1 - Experiment 1:**

Design the food blog webpage using only HTML elements

The webpage is divided into several sections

* Hyper Link 1: "Welcome to My Food Blog" - This is the main title of the blog.
* Hyper Link 2: "About Me" - This section introduces the blog author. Place the content about the author.
* List 1: "Latest Posts" - This section lists the most recent blog posts.
* List 2: "Popular Recipes" - This section lists popular recipes with links.
* Hyper Link 3: "Contact Me" - This section provides a way for visitors to contact the author.
* Hyper Link 4: "Subscribe to My Blog" - This section includes a HTML form for visitors to subscribe to the blog.

**Requirements to implement**

**1: Update Content**

Update the "Latest Posts" section by adding a new blog post titled "Healthy Salads for Every Day". Provide a brief description and a link to "salad-recipes.html".`

**2: Add a New Section**

Add a new section titled "Upcoming Events" after the "Subscribe to My Blog" section. List at least two events with their descriptions.

**3: Create a Link**

Create a hyperlink in the "About Me" section that links to the homepage of the blog. Use the text "Back to Home".

**4: Update Form Elements**

Modify the subscription form so that the email field is required and includes basic validation for email format.

**Week 2 - Experiment 2:**

Upgrade the food blog website using HTML5. The website should contain multiple sections, including an introduction about the blog author, a list of the latest blog posts, a collection of popular recipes, and a contact form for users to subscribe to the blog. The website should be user-friendly and accessible, with clear navigation and properly structured content.

**Requirements to implement:**

1. **Header Section:**
   * Create a header with a title for the blog and a navigation menu that links to different sections of the page.
   * The navigation menu should include links to the "About Me," "Latest Posts," "Popular Recipes," "Contact Me," and "Subscribe" sections.
2. **About Me Section:**
   * Include a section that provides a brief introduction to the blog author, explaining their passion for food and cooking.
3. **Latest Posts Section:**
   * Display the latest blog posts with a title, short description, and a "Read more" link that leads to the full post.
   * Include at least two blog posts in this section.
4. **Popular Recipes Section:**
   * Create a list of popular recipes with links to individual recipe pages. Each recipe should be displayed as a list item.
5. **Contact and Subscription Section:**
   * Provide a section where users can contact the blog author via email. Include a mail to link for easy access.
   * Include a subscription form where users can enter their name and email address to subscribe to the blog. The form should validate that both fields are filled out before submission.
6. **Footer Section:**
   * Add a footer that contains a copyright notice with the current year.

**Additional Information:**

* Use semantic HTML5 elements such as <header>, <section>, <article>, and <footer> to improve the structure and accessibility of the content.

**Week 3 – Experiment 3:**

Upgrade the food blog website designed in experiment 2 with CSS. Apply external styling approach.

**Requirements to apply styles:**

** Header:**

The header includes the blog's title and a navigation bar. The navigation links allow users to jump to different sections of the page. Set the background colour to light blue., keep on scrolling the header content left to right.

** Main Content:**

The main content is divided into sections (<section>) for "About the Author," "Latest Blog Posts," "Popular Recipes," and "Contact." Each section is styled with padding, a white background, rounded corners, apply element and id, class selectors.

 **Responsive Design**:

Apply media queries on popular recipes. On larger screens, the content is arranged in a grid layout, with two columns for better use of space. The grid adjusts as the screen width increases, maintaining readability and usability on various devices.

** Footer:**

The footer is simple, centered, and includes copyright information. Beautify the content with setting the margin, border style.

**Week 4 – Experiment 4:**

Upgrade food blog website structure using HTML5 and Bootstrap.

**Requirements:**

** Header:**

Uses a Bootstrap navbar for responsive navigation. The navbar toggles into a hamburger menu on smaller screens.

 **Main Content**:

**About Section**: Introduces the blog author using Bootstrap's grid system.

**Latest Blog Posts**: A row of cards (card class) to display recent posts. Bootstrap's grid system with three columns on medium and larger screens and a stacked layout on smaller screens.

**Popular Recipes**: Similar to the blog posts section, it uses cards for displaying popular recipes.

**Contact Section**: A simple form with validation using Bootstrap's form controls (form-group, form-control, and btn classes). The validation feedback is customized with Bootstrap's was-validated class.

** Footer:**

Simple and centered, providing basic copyright information.

**Week 5 – Experiment 5:**

1. Write a JavaScript program which accepts a string as input and swap the case of each character. For example, if you input 'The Quick Brown Fox' the output should be 'tHEqUICKbROWNfOX'.
2. Write a JavaScript program to find the most frequent item of an array.
3. Write a JavaScript program to remove duplicate items from an array

**Week 6 – Experiment 6:**

1. Write a JavaScript program to perform a binary search.
2. Write a JavaScript program to list the properties of a JavaScript object
3. Write a JavaScript function to check whether an object contains given property.
4. Write a JavaScript program to sort a list of elements using Quick sort.

**Week 7 – Experiment 7:**

1. Implement a basic user authentication system using Map. Each user has a unique username (as key) and a password (as value). Write functions to:

1. Register a new user.
2. Delete a user.
3. Update a user’s password.
4. Authenticate a user by checking if a username-password combination exists.
5. List all registered usernames using an iterator.

2. Write a JavaScript program to implement Bubble Sort.

**Week 8 – Experiment 8:**

1. Write a JS program to read from a JSON object and display the data in a table (HTML page).

2. Implement a user registration system where, after a user registers, a confirmation email is sent. Write functions to:

1. Register a user using a Promise that simulates an API call.
2. Send a confirmation email after successful registration using another Promise.
3. Handle errors if the registration fails or if sending the email fails.
4. Use Promise. All to register multiple users concurrently and send confirmation emails for each.

**Week 9 – Experiment 9:**

1. Write a Node JS program that accepts a port from the user and runs a node server at that port
2. Write a NodeJS program to read from a file and display the content on screen
3. Write a NodeJS program to accept a file name from user, text from user, if file exists append the text to the file. If not create a new file and add the text to it.

**Week 10 – Experiment 10:**

1. Create a student database in Mongo DB with all the details of students of a class.
2. Create a form such that, based on student roll number provided by user, the student details should be fetched (using ExpressJS)

**Week 11 – Experiment 11:**

Create a form such that CRUD operations can be performed on the student DB using ExpressJS