Programming Course

Day 1

Road map of the course.
What are front-end and back-end?
What will be the editor and development environment?
Start with HTML.

HTML Elements

1. Starting tag
2. Ending tag
3. Content between tags

Doctype

1. What are doctypes?

2. What doctype is used for in HTML?

Fundamental HTML Tags:

1. HTML tag
2. Head tag
3. Title tag
4. Body tag

Headings

1. How to create a heading in HTML?
2. How many headings are in HTML?
3. Why headings are essential for SEO?
4. Where not to use headings?

Paragraphs

1. <p> - creates paragraphs.
2. Use <hr> tag in paragraphs.
3. Use <br> tag in paragraphs.
4. Use of <pre> tag.

Text Formatting

1. <b> - Bold text
2. <strong> - Important text
3. <i> - Italic text
4. <em> - Emphasized text
5. <u> - Underline text
6. <ins> - Inserted text
7. <sub> - Subscript text
8. <sup> - Superscript text
9. <small> - Smaller text
10. <big> - Bigger text
11. <mark> - Marked text
12. <del> - Deleted text

Comments

1. <!-- - Starting tag of comment.
2. --> - Ending tag of comment.
3. How to add comments inside a paragraph & why?

Day 2

HTML Attributes

1. What are HTML attributes?

Anchors and Hyperlinks

1. <a> - link tag.
2. href - link to.
3. target - where to open the linked document.
4. _self - Default. Opens the document in the same window/tab as it was clicked.
5. _blank - Opens the document in a new window or tab.
6. Use mailto: inside the href.
7. Use tel: inside the href.

Lists

1. <ul> - unordered list.
2. <li> - list item.
3. <ol> - order list.
4. <dl> - description list.
5. <dt> - defines the term.
6. <dd> - tag describes each term.

Images

1. <img> - element to define an image.
2. src - attribute to define the URL of the image.
3. alt - attribute to define an alternate text for an image, if it cannot be displayed.
4. width and height - attributes to define the size of the image.
5. float - property to let the image float to the left or the right.
6. <picture> - element gives web developers more flexibility in specifying image resources.
7. Why not use large images on web pages?

Day 3

Tables

1. <table> - Defines a table.
2. <tr> - Defines a row in a table.
3. <th> - Defines a header cell in a table.
4. <td> - Defines a cell table.
5. <caption> - Defines a table caption.
6. <colgroup> - Specifies a group of one or more columns in a table for formatting.
7. <col> - Specifies column properties for each column within a <colgroup> element.
8. <thead> - Groups the header content in a table.
9. <tbody> - Groups the body content in a table.

10. <tfoot> - Groups the footer content in a table.

Day 4

Forms

1. <form> - element is used to create an HTML form for user input.
2. <input> - element is the most used form element.
3. <input> element can be displayed in many ways, depending on the type attribute.
4. <input type="text"> - Displays a single-line text input field.
5. <input type="radio"> - Displays a radio button (for selecting one of many choices).
6. <input type="checkbox"> - Displays a checkbox (for selecting zero or more of many choices).
7. <input type="submit"> - Displays a submit button (for submitting the form).
8. <input type="button"> - Displays a clickable button.
9. <label> - tag defines a label for many form elements.
10. <textarea> - Defines a multiline input control (text area).
11. <select> - Defines a drop-down list.
12. <option> - Defines an option in a drop-down list.
13. <optgroup> - Defines a group of related options in a drop-down list.
14. <button> - Defines a clickable button.
15. <fieldset> - Groups related elements in a form.
16. <legend> - Defines a caption for a <fieldset> element.
17. <datalist> - Specifies a list of pre-defined options for input controls.
18. <output> - Defines the result of a calculation.
19. for - attribute of the <label> tag.
20. action - attribute defines the action to be performed when the form is submitted.
21. target - works the same as <a> attribute target.
22. method - attribute specifies the HTTP method for submitting the form data. (GET or POST more on that later).
23. autocomplete - attribute specifies whether a form should have to autocomplete on or off.
24. novalidate - form attribute when present, specifies that the form-data (input) should not be validated when submitted.
25. selected - attribute to the option element.
26. multiple - attribute to the option element.

Different attributes for the HTML <input> element.

1. value - attribute specifies an initial value for an input field.
2. readonly - attribute specifies that an input field is read-only.
3. disabled - attribute specifies that an input field should be disabled.
4. size - attribute specifies the visible width, in characters, of an input field.

5. maxlenght - attribute specifies the maximum number of characters allowed in an input field.
6. min and max attributes specify the minimum and maximum values for an input field.
7. multiple - attribute specifies that the user can enter more than one value in an input field.
8. pattern - attribute specifies a regular expression that the input field's value is checked against when the form is submitted.
9. placeholder - attribute specifies a short hint that describes the expected value of an input field (a sample value or a short description of the expected format).
10. required - attribute specifies that an input field must be filled out before submitting the form.
11. step - attribute specifies the legal number intervals for an input field.
12. autofocus - attribute specifies that an input field should automatically get focused when the page loads.

Day 5

Div Element

1. <div> - tag defines a division or a section in an HTML document.
2. <div> - tag is used as a container for HTML elements
3. <div> - tag is easily styled using the class or id attribute.
4. Any sort of content can be put inside the <div> tag!

Sectioning Elements

1. What are sectioning elements?
2. <article> - Defines independent, self-contained content.
3. <aside> - Defines content aside from the page content.
4. <details> - Defines additional details that the user can view or hide.
5. <figcaption> - Defines a caption for a <figure> element.
6. <figure> - Specifies self-contained content, like illustrations, diagrams, photos, code listings, etc.
7. <footer> - Defines a footer for a document or section.
8. <header> - Specifies a header for a document or section
9. <main> - Specifies the main content of a document.
10. <mark> - Defines marked/highlighted text.
11. <nav> - Defines navigation links.
12. <section> - Defines a section in a document.
13. <summary> - Defines a visible heading for a <details> element.
14. <time> - Defines a date/time.

Meta Information

1. <meta> - tag defines metadata about an HTML document.

2. Why do we use meta tags?
3. <meta> tag has the following attributes.
4. charset - specifies the character encoding for the HTML document.
5. content - Specifies the value associated with the http-equiv or name attribute.
6. http-equiv - Provides an HTTP header for the information/value of the content attribute.
7. name - Specifies a name for the metadata.
8. The name attribute has the following properties.
9. keywords, description, author, refresh, viewport.

CSS Basics

1. What is CSS?
2. CSS Syntax.
3. CSS Selectors.
4. How to add CSS?
5. CSS comments
6. CSS selectors precedence.
7. What is important?

Text Formatting

1. Text Alignment
    1.1. text-align
    1.2. text-align-last
2. Text Colors
    2.1. color
3. Text Decoration
    3.1. text-decoration-line
    3.2. text-decoration-color
    3.3. text-decoration-style
    3.4. text-decoration-thickness
    3.5. text-decoration
4. Text Transformation
    4.1. text-transform
5. Text Spacing
    5.1. text-indent
    5.2. letter-spacing
    5.3. line-height
    5.4. word-spacing
    5.5. white-space
6. Text Shadow
    6.1. text-shadow

Day 6

CSS Font

1. Family
    1.1. font-family
2. Size
    2.1. font-size
3. Style
    3.1. font-style
4. Weight
    4.1. font-weight

Box Model

1. CSS Border.
2. CSS Margins.
3. CSS Padding.
4. CSS Height and Width.
5. CSS Box Shadow.

Day 7

CSS Backgrounds

1. Background color
2. Background image
3. Properties:
    3.1. background-color
    3.2. background-image
    3.3. background-size
    3.4. background-repeat
    3.5. background-position
    3.6. background-clip

CSS Layout

1. CSS Overflow.
2. CSS Units.
    2.1. px
    2.2. em
    2.3. rem
    2.4. vh
    2.5. vw
    2.6. %

Day 8

CSS Layout

1. CSS Position
    1.1. Static Position
    1.2. Relative Position
    1.3. Fixed Position
    1.4. Absolute Position
    1.5. Sticky Position
2. CSS Z-index.

Day 9

CSS Layout

1. CSS Display
    1.1. Inline
    1.2. Block
    1.3. Inline-block
    1.4. None
2. CSS Visibility
    2.1. Visibile
    2.2. Hidden

Day 10

CSS Flexbox

1. What is CSS Flexbox?
2. What are flex components?
3. Flex container properties.
    3.1. flex-direction
    3.2. flex-wrap
    3.3. flex-flow
    3.4. justify-content
    3.5. align-items
    3.6. align-content
4. Flex item properties.
    4.1. order
    4.2. flex-grow
    4.3. flex-shrink
    4.4. flex-basis

    4.5. flex
    4.6. align-self

Day 11

CSS Grid

5. What is CSS Grid?
6. What is the difference between grid and flex?
7. What are grid components?
8. Grid container properties.
    8.1. grid-template-columns
        8.1.1. px, fr, auto
        8.1.2. repeat(), minmax(), auto-fit
    8.2. grid-template-rows
        8.2.1. px, auto
    8.3. grid-auto-columns
    8.4. grid-auto-rows
    8.5. column-gap
    8.6. row-gap
    8.7. gap
    8.8. justify-content
    8.9. align-content
    8.10. grid-template-area
9. Grid item properties.
    9.1. grid-column-start
    9.2. grid-column-end
    9.3. grid-column
    9.4. grid-row-start
    9.5. grid-row-end
    9.6. grid-row
    9.7. grid-Area

Day 12

CSS Media Queries

1. What is responsive web design?
2. What are media queries?
3. Why are media queries used?
4. What are media query breakpoints?

5. Typical media query breakpoints
    5.1. @media only screen and (max-width: 600px) [Extra small devices (phones, 600px and down)]
    5.2. @media only screen and (min-width: 600px) [Small devices (portrait tablets and large phones, 600px and up)]
    5.3. @media only screen and (min-width: 768px) [Medium devices (landscape tablets, 768px and up)]
    5.4. @media only screen and (min-width: 992px) [Large devices (laptops/desktops, 992px and up)]
    5.5. @media only screen and (min-width: 1200px) [Extra large devices (large laptops and desktops, 1200px and up)]

Day 13

Final Project: Create a fully responsive page using HTML and CSS.

Day 14

Getting Started with JavaScript

1. What is JavaScript?
2. What can you do using JavaScript?
3. What is the difference between JavaScript and HTML?
4. Different ways of using JavaScript with HTML?
5. What are JavaScript comments?
6. What are the different data types?
    6.1. String
    6.2. Integer
    6.3. Float
    6.4. Boolean
    6.5. Array
7. What are variables?
8. Different ways to print values in JavaScript.
    8.1. console.log()
    8.2. alert()
    8.3. document.write()
9. What are JavaScript assignment operators?
10. What are arithmetic operators?

Task

1. Find the area of a rectangle where the length is 5 and the width is 8.
2. Find the area of a triangle where the base is 4 and the height is 3.

3. Find the area of a circle where the radius is 3.
4. Convert temperatures from Celsius to Fahrenheit and Fahrenheit to Celsius.

Day 15

JavaScript Conditional Operators

1. < less than
2. > greater than
3. <= less than or equal to
4. >= greater than or equal to
5. == equal to
6. === identical to
7. != not equal to
8. !== not identical to

JavaScript Logical Operators

1. && AND
2. || OR
3. ! NOT

JavaScript Conditional Statement

1. Use **if** to specify a block of code to be executed, if a specified condition is true.
2. Use **else** to specify a block of code to be executed if the same condition is false.
3. Use **else if** to specify a new condition to test, if the first condition is false.

Task

1. Check two given numbers and return true if one of the numbers is 50 or if their sum is 50.
2. Check from the given integer, whether it is positive or negative.
3. Check whether a given number is even or odd.
4. Check whether a given positive number is a multiple of 3.
5. Determine whether a given year is a leap year.
6. Check from the given two numbers, whether the first number is "greater", "lesser" or "equal" to the second number.
7. Check from the three sides of the triangle. use conditions to determine and display weather the triangle is "Equilateral" (all sides are equal), "Isosceles" (two sides are equal), or "Scalene" (no sides are equal)
8. Check from the given month (1-12) if it's "Winter" (December-February), "Spring" (March-May), "Summer" (June-August), or "Autumn" or "Fall" (September-November).

Day 16

JavaScript Functions

1. What are JavaScript Functions?
2. Why make JavaScript functions?
3. How to make JavaScript functions?
4. What is function scope?

5. How to call JavaScript functions?
6. What are function arguments/parameters?
7. What are argument default values and how to set them?
8. What are normal/regular functions ?

Tasks

1. All the prior tasks are to be solved with functions.

Day 17

JavaScript Arrow Functions

1. What are anonymous functions?
2. How to make anonymous functions?
3. How to call anonymous functions?
4. What are arrow functions?
5. How to make arrow functions?
6. How to call the arrow function?

Task

1. All the prior tasks are to be solved with arrow functions.

Day 18

JavaScript Arrays

1. What are JavaScript Arrays?
2. Why use JavaScript Arrays?
3. How to make an array?
4. How to access array elements?
5. How to Recognize an Array?
    - 5.1. typeof

    5.2. Array.isArray()
    5.3. array instanceof Array
6. Few Array methods:
    6.1. toString()
    6.2. join()
    6.3. push()
    6.4. pop()
    6.5. unshift()
    6.6. shift()
    6.7. splice()
    6.8. slice()
    6.9. Delete array[index]
    6.10. concat()
    6.11. indexOf()
    6.12. lastIndexOf()
    6.13. includes()

Task

1. Find the last element of an array without giving a hard-coded index.
2. Check whether the first or the last index of an array has a specified value, let's say 5.
3. Make an array to store the names of students and check whether that array has your own name or not and also return the index of that value.
4. Add the array element at the specified index.
5. Delete the array element from the specified index.

Day 19

JavaScript Objects

1. What is the difference between an array and an object?
2. What are JavaScript Objects?
3. Why use JavaScript Objects?
4. How to make an object?
5. How to access object properties?

Task

1. Make an array of objects (an array that will have objects) to store the student information and access it as hard-coded.

   Information to store:

Name, age, registration number, course, favorite programming languages (should be an array), Marks of 5 different subjects (should be an object).

The operations to perform:

Display every single property of any student.

Display the first and last favorite programming language of any student.

Display the marks of any 2 subjects of any student.

Day 20

JavaScript Loops

1. What are JavaScript Loops ?
2. What is a **For** loop ?
3. What is a **For In** loop ?
4. What is a **For Of** loop ?
5. What is a **While** loop ?
6. What is a **Do While** loop ?
7. What is a **Foreach** loop ?

Task

1. Sum all the array elements by using a loop.
2. Make a reverse of the array by using a loop.
3. Find the largest number in an array by using a loop.
4. Find the smallest number in an array by using a loop.
5. Write a function that checks if a word is a palindrome (reads the same forwards and backward).
6. Write a function to calculate the factorial of a given number.
7. Write a function that determines whether a given number is prime or not.
8. Print numbers from 1 to 100, but for multiples of 3 print "Fizz", for multiples of 5, print "Buzz" and for numbers that are multiples of both 3 and 5 print "FizzBuzz".
9. Find the same numbers from the following multidimensional array

```
let multiArr = [

  [

    [12, 32, 13, 34],

    [13, 12, 23, 41],

    [15, 23, 34, 45],
  ],
  [
    [122, 32, 133, 314],
    [123, 132, 23, 141],
    [155, 23, 334, 465],
  ],
  [
    [12, 342, 135, 234],
    [713, 712, 423, 431],
    [15, 23, 34, 45],
  ],
  [
    [12, 372, 913, 334],
    [13, 162, 243, 341],
    [175, 423, 34, 435],
  ],
];
```

Day 21

DOM

1. What is DOM?
2. What are DOM methods and properties?
3. How to find HTML elements by getElementById?
4. How to find HTML elements by getElementsByTagName?
5. How to find HTML elements by getElementsByClassName?
6. How to find HTML elements by querySelector?
7. How to find HTML elements by querySelectorAll?
8. How to get & change the content of HTML elements?
    a. innerText
    b. innerHTML

9. How to change the style (CSS) of HTML elements?
    a. style
10. How to add HTML elements?
    a. createElement()
    b. append()
    c. prepend()
11. How to delete HTML elements?
    a. remove()
12. How to add, change, and delete the HTML attributes?
    a. getAtrribute()
    b. setAtrribute()
    c. attrbuteName (as property)
    d. removeAttribute()
13. How to play with Data Attributes?
    a. .dataset
14. How to play with Element Classes?
    a. .classList
    b. .add()
    c. .remove()
    d. .toggle()

Day 22

JS Event Listeners

1. What are JS event listeners and how to use them?
2. How to add an event listener?
    2.1. AddEventListener()
3. What are different events?
    3.1. click
    3.2. contextMenu
    3.3. mouseOver
    3.4. mouseOut
    3.5. submit
    3.6. focus
4. How to add more than one event listener ?
5. How to run an event listener for once only ?
6. How to stop event propagation?
    6.1. stopPropagation()

Task

1. Create a button that changes its background color when clicked.
2. Build a counter that increments when a button is clicked and displays the count on the page.
3. Create a list of items. implement a way to remove an item when it is clicked.
4. Create a button that enlarges the image when clicked.
5. When a button is clicked, display a random interesting fact from an array.

Day 23

Form Validation

1. What is form validation ?
2. Why is form validation important ?
3. How to perform form validation ?

Task

1. Create a form with a dropdown menu for selecting a country. depending on the selected country, dynamically show or hide fields that are specific to that country's address format. For example, if "United States" is selected, show fields for state and ZIP code; if "Canada" is selected, show fields for province and postal code.

Day 24

DOM Practice

Day 25

Final Project 1: .Create a JavaScript QuizApp

Day 26

QuizApp Practice.

Day 27

Final Project 2: Create a JavaScript Todo App.

Day 28

Todo App Practice.

Day 29

Final Project 3: Create a Weather API application.

Day 30

Weather App Practice.

Day 31

Getting Started with Bootstrap

1. What is Bootstrap?
2. Containers and Breakpoints.
    - 2.1. .container
    - 2.2. .container-sm
    - 2.3. .container-md
    - 2.4. .container-lg
    - 2.5. .container-xl
    - 2.6. .container-xxl
    - 2.7. .container-fluid
3. Columns.
    - 3.1. .col-
    - 3.2. .col-sm-
    - 3.3. .col-md-
    - 3.4. .col-lg-
    - 3.5. .col-xl-
    - 3.6. .col-xxl-
4. Rows.
    - 4.1. .row-
    - 4.2. .row-cols-
    - 4.3. .row-cols--sm-
    - 4.4. .row-cols--md-
    - 4.5. .row-cols--lg-
    - 4.6. .row-cols--xl-

4.7. .row-cols--xxl-
4.8. .g-
4.9. .gx-
4.10. .gy-

5. Typography.
   5.1. Heading classes (.h1, .h2, .h3, .h4, .h5, and .h6).
   5.2. Display classes (.display-1, .display-2, .display-3, .display-4, .display-5, and .display-6).
   5.3. Text alignment (.text-start, .text-center, .text-end).
   5.4. Others (.lead, .text-muted, .small, .mark, etc.).
6. Tables.
   6.1. .table
   6.2. .table-{colour}
   6.3. .table-striped
   6.4. .table-hover
   6.5. .table-responsive
7. Forms.
   7.1. .form-control
   7.2. .form-select
   7.3. .form-control-color
   7.4. .form-check
   7.5. .form-check-input
   7.6. .form-check-label
   7.7. .form-switch
   7.8. .input-group
   7.9. .input-group-text
8. Buttons.
   8.1. .btn
   8.2. .btn-{colour}
   8.3. .btn-outline-{colour}
9. Alerts.
   9.1. .alert
   9.2. .alert-{colour}
   9.3. .alert-link
10. Cards.
    10.1. .card
    10.2. .card-header
    10.3. .card-body
    10.4. .card-title
    10.5. .card-text

    10.6. .card-footer
11. Modals.
    11.1. .modal
    11.2. .modal-dialog
    11.3. .modal-content
    11.4. .modal-header
    11.5. .modal-body
    11.6. .modal-footer
12. Navbars.
13. Color Utilities.
    13.1. .bg-{colour}
    13.2. .text-{colour}
    13.3. .text-bg-{colour}
14. Border Utilities.
    14.1. .border
    14.2. .border-{colour}
    14.3. .border-{side}
    14.4. .rounded
15. Display Utilities.
    15.1. .d-none
    15.2. .d-block
    15.3. .d-{breakpoint}-none
    15.4. .d-{breakpoint}-block
16. Spacing Utilities.
    16.1. .m-
    16.2. .m{side}-
    16.3. .p-
    16.4. .p{side}-

Day 32

Getting Started with PHP

1. What is PHP & what can it do?
2. What is a serverside and clientside programming language ?
3. How to Install PHP on a Computer?
4. How does PHP work with HTML?
5. PHP basic Syntax.
6. PHP Comments.
7. PHP different data types.

    a. String
    b. Integer
    c. Float
    d. Boolean
    e. Array
8. PHP Variables.
9. PHP Constants.
10. Different ways to print strings in PHP on screen.
    a. echo
    b. print
    c. print_r()
    d. var_dump()
11. What are PHP assignment operators?
12. What are arithmetic operators?

Task

1. Find the area of a rectangle where the length is 5 and the width is 8.
2. Find the area of a triangle where the base is 4 and the height is 3.
3. Find the area of a circle where the radius is 3.
4. Convert temperatures from Celsius to Fahrenheit and Fahrenheit to Celsius.

Day 33

PHP Conditional Operators

1. < less than
2. > greater than
3. <= less than or equal to
4. >= greater than or equal to
5. == equal to
6. === identical to
7. != not equal to
8. !== not identical to

PHP Logical Operators

4. && AND
5. || OR
6. ! NOT

PHP Conditional Statement

4. Use **if** to specify a block of code to be executed, if a specified condition is true.
5. Use **else** to specify a block of code to be executed if the same condition is false.
6. Use **else if** to specify a new condition to test, if the first condition is false.

PHP Functions

1. What are PHP Functions?
2. Why make PHP functions?
3. How to make PHP functions?
4. What is function scope?
5. What are function arguments/parameters?
6. What are argument default values and how to set them?
7. How to call PHP functions?
8. What are anonymous functions?
9. How to make anonymous functions?
10. How to call anonymous functions?
11. What are arrow functions?
12. How to make arrow functions?
13. How to call the arrow function?

Task

1. Check two given numbers and return true if one of the numbers is 50 or if their sum is 50.
2. Check from the given integer, whether it is positive or negative.
3. Check whether a given number is even or odd.
4. Check whether a given positive number is a multiple of 3.
5. Determine whether a given year is a leap year.
6. All the prior tasks are to be solved with functions.

Day 34

PHP Arrays

1. What are PHP Arrays?
2. Why use PHP Arrays?
3. How to make an array in PHP?
4. How to access array elements?
5. How to Recognize an Array?
   5.1. is_array()
6. What is a Normal Array?
7. What is an Associative Array?
8. Few Array methods:
   8.1. count()

8.2. in_array()
8.3. array_search()
8.4. array_push()
8.5. array_pop()
8.6. array_unshift()
8.7. array_shift()
8.8. array_slice()
8.9. array_splice()
8.10. array_merge()
8.11. array_combine()
8.12. array_keys()
8.13. array_values()
8.14. array_flip()
8.15. array_reverse()

PHP Loops

1. What are PHP Loops?
2. Why use PHP Loops?
3. What is a **for** loop?
4. What is a **while** loop?
5. What is a **do-while** loop?
6. What is a **foreach** loop?

Task

1. Sum all the array elements by using both a loop and a PHP built-in function.
2. Reverse the array by using both a loop and a PHP built-in function.
3. Make a table of the given number with all possible loops.
4. Find the largest number in an array by using both a loop and a PHP built-in function.
5. Find the smallest number in an array by using both a loop and a PHP built-in function.
6. Add the array element at the specified index.
7. Delete the array element from the specified index.
8. Make a normal array to store the name of 5 students and iterate with for and for each loop.
9. Make an associative array to store the name of 5 students and iterate with a for each loop.
10. Make a normal array of associative arrays(an array that will have associative arrays) to store the information of at least 2 students and access them as hard-coded.

    Information to store:

    Name, age, registration number, course, favorite programming languages (should be a normal array), Marks of 5 different subjects (should be an associative array).

The operations to perform:

Display every single value for any student.

Display the first and last favorite programming languages of any student.

Display the marks of any two subjects for any student.

Day 35

PHP form handling and validation

1. What is form handling in PHP?
2. What are $_SERVER, $_POST, and $_GET?
3. What is form validation?
4. What is input sensitization?
5. Why is input sensitization important?
6. Why is server-side form validation important?

Tasks

1. Make a form to collect the details of a student with proper validation and sanitization.
2. Make a form to perform arithmetic operators with proper validation and sanitization.
3. Make a form to perform other operations that were described in class with proper validation and sanitization.

Day 36

PHP file uploading

1. How are files uploaded in PHP?
2. How to validate files in PHP?
3. What is $_FILES?
4. How are files stored in PHP?

Task

1. Make a form to collect the details of a student with a profile picture and perform proper validation and sanitization.

Day 37

MySQL Database

1. What is a MySQL Database?

2. When to use MySQL Database?
3. Where does it run?
4. How to open PHPMyAdmin?
5. How to create a database in PHPMyAdmin?
6. How to create a table in a database?
7. How to export and import database files in PHPMyAdmin?
8. What are basic database queries?
9. What is the **where** clause?

Task

1. Create a table of users and perform database queries.

Day 38

MySQL Database

1. How to connect a database in PHP?
2. How to perform CRUD (Create, Read, Update, and Delete) in PHP & MySQL?
3. What is git?
4. How to upload a project on git?
5. Git commands for uploading:
    - 5.1. git init
    - 5.2. git add .
    - 5.3. git commit -m "commit message"
    - 5.4. git remote add origin https://github.com/username/repo-name.git
    - 5.5. git push -u origin master
6. How to clone a project from git?
7. Git commands for cloning:
    - 7.1. git clone https://github.com/username/repo-name.git

Task

1. Perform CRUD through an interface on users' data properly.

Day 39

Project 1: Create a PHP login and CRUD.

Day 40

The practice of CRUD and login systems.

Day 41

Project 2: Create a PHP Async CRUD.

Day 42

The practice of PHP async CRUD.

Day 43

Project 3: Create a PHP Todo app with authentication.

Day 44

The practice of PHP Todo app with authentication.

Day 45

Project 4: Create a PHP Student Registration System.

Day 46

The practice of the PHP Student Registration System.

Day 47

OOP PHP

1. What is a class?
2. How to make a class?
3. What are class properties and methods?
4. What are access modifiers?
5. What is an object?
6. How to make an object?
7. How to access properties and methods?
8. What is a constructor?
9. What is a destructor?
10. What is inheritance?

Day 48

OOP PHP

1. What are static properties and methods?
2. How to access static properties and methods?
3. Abstract Class
4. Interfaces

Day 49

OOP PHP

1. Traits
2. Namespaces

Day 50

Project: How to perform CRUD (Create, Read, Update, and Delete) in OOP PHP & MySQL?

Day 51

Getting started with Laravel

1. What tools are required for Laravel?
2. What is Laravel?
3. What is MVC Architecture?
4. How to set up a Laravel app?
5. What are Routes in Laravel?
6. What is an artisan in Laravel?
7. How to list all commands?
8. How to create a Route?
9. How to list routes?
10. What is a blade template?
11. How to call View from Route?
12. How to send data (variables, arrays) from routes to views?
13. How to receive data (variables, arrays) in views?
14. What are Blade Directives?

Day 52

Laravel Controller

1. What is a Controller in Laravel?
2. How to create a controller?
3. Why do we need a controller?

4. How to create a resource controller?
5. How to create a resource route?
6. How to connect routes, controllers, and views?
7. How to send data (variables, arrays) from the controllers to views?

Day 53

Laravel Migrations

1. What are Laravel migrations?
2. Why do we need migrations?
3. How significant are migration names?
4. How to create a migration?
5. What are migration Commands?
    - 5.1. php artisan make:migration {name}
    - 5.2. php artisan migrate
    - 5.3. php artisan migrate:rollback
    - 5.4. php artisan migrate:status
    - 5.5. php artisan migrate:refresh
    - 5.6. php artisan migrate:reset
6. How to create a new table?
7. How to add a column to an existing table?

Task

1. How to drop a column from a table?
2. How to rename an existing column?
3. How to rename an existing table?
4. How to change the datatype of a column?
5. How to drop a table?

Day 54

Laravel Models

1. What are Laravel models?
2. How to create a model by using the artisan command?
    - 2.1. Model alone.
    - 2.2. Model and migration.
    - 2.3. Model, migration, and controller.
    - 2.4. Model, migration, and resource controller.
3. What is model fillable?
4. What is mass assignment?

CRUD with Laravel model

1. Perform a CRUD by using the model with UI.

Laravel Factories and Seeders

1. What are Laravel factories?
2. How to create a factory?
3. Why do we need factories?
4. How to create a seeder?
5. How to seed the database from factories?
6. How to seed the database from factories without migrations?

Day 55

Laravel Soft deletes

1. What are Laravel soft deletes?
2. How to add soft delete to new and existing models?
3. How to activate soft delete in a model?
4. How to get trashed records?
5. How to get all records?
6. How to recover trashed records?
7. How to forcefully delete records?

Day 56

Laravel authentication system

Custom:

1. Login view
2. Register view
3. Logout
4. Authentication controller
5. Routes grouping and name
6. Middleware

Laravel Breeze:

1. What is Laravel Breeze?
2. How to use it?
3. How to install it?
   a. composer require laravel/breeze
   b. php artisan breeze:install

    c. php artisan migrate
    d. npm install
    e. npm run dev
4. How to customize it?

Days 57, 58, 59, 60, & 61

A sample laravel project (Contact App) to demonstrate the following.

1. Template breakdown
2. Authentication
3. Relationships
4. Pagination
5. Middleware
6. Group routes
7. Routes naming
8. File uploading
9. Blade components

From day 62 to the end of the course

Make a big project like a Learning Management System, Customer Relation Management, Inventory Management System, Employee Management System , Point Of Sale, Etc.