Web Development Using PHP

Assignment No.1

Category 1: Arrays & Basic PHP Operations

1. WAP using PHP that will perform the following operations on an array & display the output on a webpage:
   - Sort an associative array in ascending order by value.
   - Sort an index array in descending order.
   - Sort an associative array in ascending order by key.
   - Sort an associative array in descending order by value.
2. Write a PHP code that will accept a name from the user & display the message "Name must have at least 4 characters" if the name length is less than 4, else display "hello ...(name)..".
3. Write code that will accept two numbers from the user & provide four buttons on the webpage (add, sub, mul, div). Depending on the user's click, perform that operation & display the result on the **same** webpage.
4. Write code that will accept two numbers from the user & provide four buttons on the webpage (add, sub, mul, div). Depending on the user's click, **redirect the user to different pages**, perform the operation there & display the result.
5. Write an HTML code that will accept a paragraph from the user; display the count of the words present in the paragraph using PHP.
6. Write an HTML code that will accept a paragraph from the user; replace every occurrence of "and" with "sand" & display the updated paragraph on the webpage using PHP.
7. Write an HTML code that will accept a paragraph from the user; replace every occurrence of "and" with "land" & display the updated paragraph on the webpage using PHP.
8. Write an HTML code that will accept a paragraph from the user; display the **first five words** present in the paragraph using PHP.
9. Write an HTML code that will accept a name and age from a student. Display the name of the student as many times as his age on the webpage & make sure that the user must not keep fields blank. (Using PHP)
10. Write a PHP code that will demonstrate the use of the following array functions: sort(), rsort(), asort(), ksort(), arsort(), krsort().

Category 2: Form Validation & User Input

11. Write code that will accept a name & age from the user. If the age is less than 18, display the message “not eligible for voting”, otherwise display “eligible for voting”.

12. Write code that will accept details of a student like name, age, gender, address, course, hobbies and display the entered information on the webpage using PHP.

13. Write code that will accept details of a student like name, age, gender, address, course, hobbies and display the entered information in a **tabular format** on the webpage using PHP.

14. Design an HTML page that will accept details of a person (name, age, gender, address, state, hobbies, favourite color, date of birth, email id, telephone number) using HTML5 controls. Compel the user to enter all information. Check if the age is above or below 18 years and display "eligible for voting" or "not eligible for voting" along with the entered details.

15. Write a PHP code that will accept old password & new password from the user; display appropriate messages if both passwords matched or unmatched.

16. Design an HTML page with two textareas and a button. Accept a paragraph from the first textarea. On the click of the button, display the accepted string in **reverse order and in upper case** in the second textarea using PHP.

17. Accept a paragraph from the user using an HTML page. Display the count of the number of lines, words, and characters on the webpage using PHP.

Category 3: Database Operations (Insert, Display, Delete, Update)

18. Design an HTML page that will accept details of an Employee (emp no, name, age, gender, address). Insert the employee details into an EMP table (assume suitable fields) **only if the employee record doesn't already exist** & display the details of the employee in tabular format on the webpage.

19. Write a PHP code that will display records of students who live in **Pune** city in a tabular format on the webpage. (Assume a suitable student table)

20. Write a PHP code that will display records of students who live in **Aurangabad** city in a tabular format on the webpage. (Assume a suitable student table)

21. Write a PHP code that will display records of students who **do NOT** live in Mumbai in a tabular format & **delete** records of students who live in Mumbai.

22. Design an HTML page that will accept a student ID from the user; **delete** the student record from the students' table & display the deleted record on the webpage using PHP.

23. Write a program (WAP) that will **update** a particular record in the student table. (Assume suitable fields)

24. Write a PHP code that will display the record of employees in **ascending order of their salary** on the webpage. (Assume an employee table with suitable fields)

25. Write code that will accept an EMP ID from a webpage. On the click of a button, **delete** the record of that employee. **Before deleting**, display the record on the webpage using PHP.

26. Write a PHP code that accepts the details of a student (name, age, gender, address, course, hobbies). **Update** the particular record of that student with the new details.

27. Write a PHP code that will **delete** records of employees whose salary is less than 20000 Rs. and display **only the number of deleted records**.

28. Write code that will accept feedback from a user in HTML & insert the accepted details into a feedback table. (Validate that no field remains blank).

29. Write a PHP code that will display elements of a **fruit table** (assume a suitable table) as **checkboxes** in HTML.

30. Write a PHP code that will display elements of a **course table** (assume a suitable table) as **radio buttons** in HTML.

31. Write an HTML code that will accept employee details (emp id, name, age, address, mobile number, gender, hobbies) from the user. Store the details in an EMP_TABLE. Before storing, ensure the EMP_TABLE has all **unique records**. If the record already exists, display a suitable message.

Web Development Using PHP

Assignment No.2

Category 6: File Handling & Uploads

32. Write a program to accept employee details along with an image file (less than 100 KB in size). Store the file in a folder “impfile” & other details in an “emp” table.
33. Write a program to accept an emp id from a webpage. On the click of a button, display the record of the employee along with its image on the webpage. (Assume the emp table already exists & images are stored in the “impfile” folder).
34. Write a program to upload **only a PDF file** (less than 100 KB in size) & store the file in a folder “impfile” **using a database**. Also, display the list of all uploaded files in **ascending order of the date of upload**.
35. Write an HTML code that will accept employee details (emp id, name, age, address, mobile number, gender, hobbies, PHOTOGRAPH) from the user. Store the details in an EMP_TABLE. Before storing, ensure the PHOTOGRAPH size is < 100 KB and the format is “jpg”/ “jpeg” / “png” only. **(Store the PHOTOGRAPH in the EMP_TABLE as a BLOB)**.
36. Write an HTML code that will accept employee details (emp id, name, age, address, mobile number, gender, hobbies, PHOTOGRAPH) from the user. Store the details in an EMP_TABLE. Before storing, ensure the PHOTOGRAPH size is < 100 KB and the format is “jpg”/ “jpeg” / “png” only. **(Store the PHOTOGRAPH in an "EMP" folder and save its path in the EMP_TABLE)**.
37. Write an HTML code that will accept employee details (emp id, name, age, address, mobile number, gender, hobbies, PHOTOGRAPH, UG Degree) from the user. Store the details in an EMP_TABLE. Before storing, ensure both files are < 100 KB, the PHOTOGRAPH format is “jpg”/ “jpeg” / “png”, and the UG Degree is in PDF format. **(Store both files in an "EMP" folder and save their paths in the EMP_TABLE)**.
38. Consider two txt files with some content. **Swap the content** of these two files using file handling.
39. Consider two txt files with some content. Read **even-numbered lines** from the first file (“first.txt”) & **odd-numbered lines** from the second file (“second.txt”) and write them into a third file (“third.txt”) such that an even line from “first.txt” is **immediately followed** by an odd line from “second.txt”.
40. Consider two txt files with some content. Read **all even-numbered lines** from the first file (“first.txt”) & **all odd-numbered lines** from the second file (“second.txt”) and write them into a third file (“third.txt”) such that **all even lines are first, followed by all odd lines**.
41. Consider the file “first.txt” with some content. Read the file and display the count of the number of lines, words, and characters on the webpage using file handling in PHP.

42. Consider a directory “category” which has several files. Search for a “first.txt” file in it. If present, open it and write a message (“Hello we have found the file”) in it. If not present, display a message (“Sorry we haven’t found the file”) on the webpage.

Category 7: OOPs Concepts

43. Write PHP codes that will implement the following concepts: Inheritance, Interface, Passing parameters to a base class constructor, Abstract class.

Category 3: Session Handling

44. Write PHP codes for the implementation of an **online watch shopping** website using session handling. (Display watches on 2-3 pages & display a bill for selected items from these pages).

45. Write PHP codes for the implementation of an **online laptop shopping** website using session handling. (Display laptops on 2-3 pages & display selected laptops in a tabular format at the end).

46. Write a PHP code that demonstrates the concept of session creation, assigning session variables, accessing variables, and destroying session variables.

Category 4: Cookies

47. Write a PHP code that demonstrates the concept of Cookies: creation, setting, accessing, and destroying.