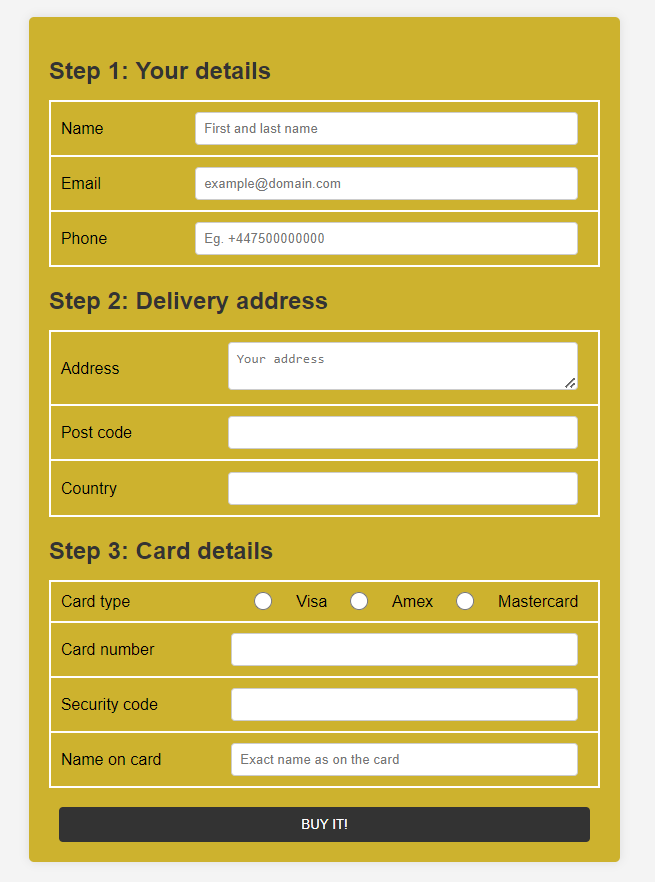
**PHP Assignment**

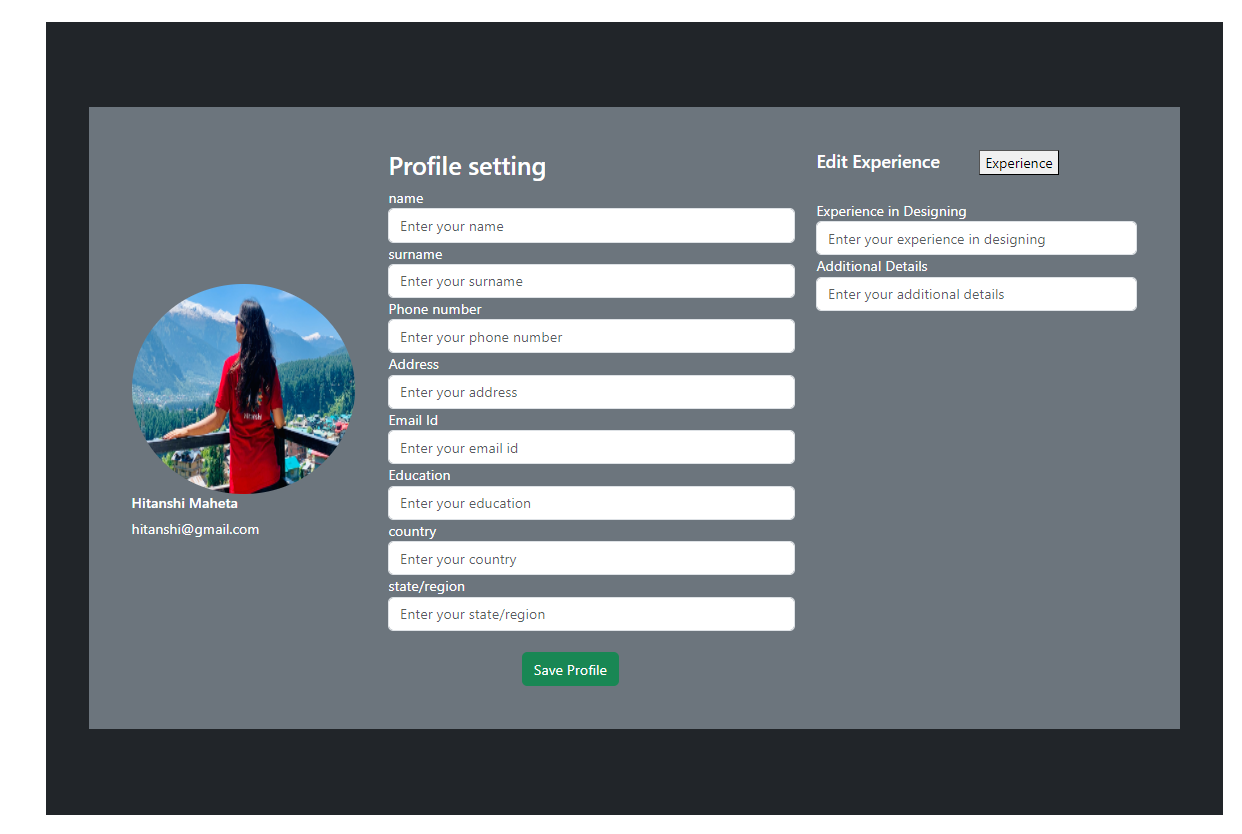
**MODULE – 2 HTML, CSS, SQL**

Exercise 1: Try To Create User Details Form.

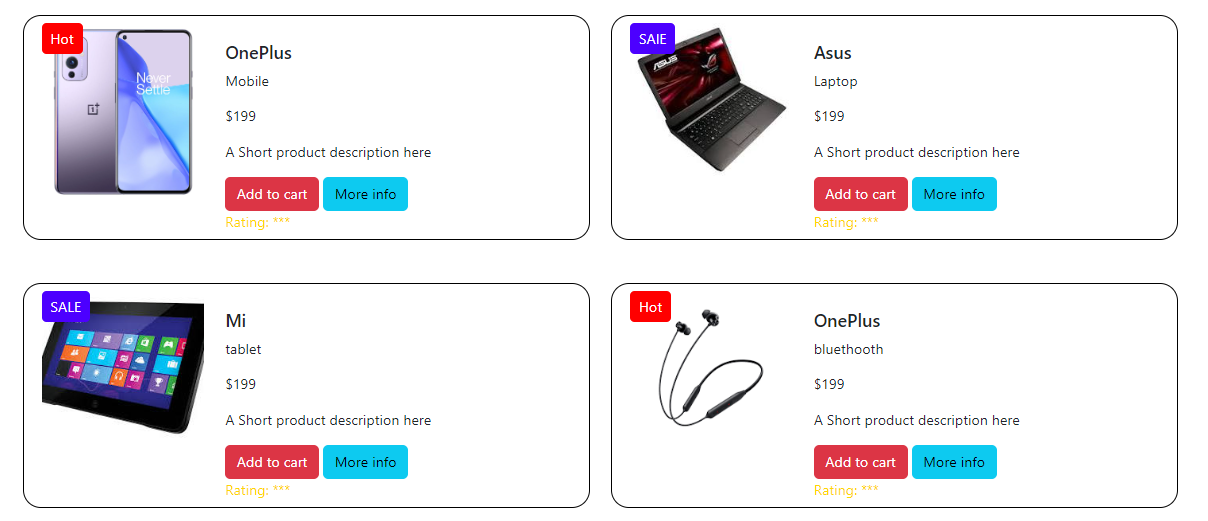


**BOOTSRAP**

Exercise 1: Create manage Profile:

****

Exercise2 : CreateProductListing Page:

****

Topic SQL

Task-1

1. Get First\_Name from employee table using alias name “Employee Name”.

Ans: SELECT First\_Name AS 'Employee Name'FROM employee;

1. Get FIRST\_NAME, Joining year, Joining Month and Joining Date from employee table.

Ans: SELECT FIRST\_NAME,

TO\_CHAR(joining\_date, 'YYYY') AS JoinYear,

TO\_CHAR(joining\_date, 'Mon') AS JoinMonth,

TO\_CHAR(joining\_date, 'dd') AS JoinDate FROM employee;

1. Get all employee details from the employee table order by First Name Ascending And Salary descending?

Ans: SELECT \*FROM employee ORDER BY First\_Name ASC, Salary DESC;

1. Get employee details from employee table whose first name contains “%o”.

Ans: SELECT \*FROM employee WHERE First\_Name LIKE 'o';

1. Get employee details from employee table whose joining month is “January”.

Ans: SELECT \*FROM employee WHERE joining month = 01;

1. Get department, total salary with respect to a department from employee table Order By total salary descending.

Ans: SELECT DEPARTMENT\_NAME, SUM(SALARY) AS Total\_Salary FROM employee GROUP BY DEPARTMENT\_NAME ORDER BY Total\_Salary DESC;

1. Get department wise maximum salary from employee table order by salary ascending?

Ans: SELECT DEPT\_ID, MAX(SALARY) FROM department GROUP BY DEPT\_ID ORDER BY MAX(SALARY) ASC;

1. Select first\_name, incentive amount from employee and incentives table for those Employees who have incentives and incentive amount greater than 3000.

Ans: SELECT FIRST\_NAME, INCENTIVES FROM employee A INNER JOIN incentives B ON A.EMPLOYEE\_ID = B.EMPLOYEE\_REF\_ID WHERE INCENTIVE\_AMOUNT > 3000;

1. Select 2nd Highest salary from employee table.

Ans: SELECT MAX(Salary) FROM Employee WHERE Salary < (SELECT MAX(Salary) FROM Employee);

1. Select first\_name, incentive amount from employee and incentives table for all Employees who got incentives using left join.

Ans: SELECT FIRST\_NAME, IFNULL(B.INCENTIVE\_AMOUNT, 0) FROM employee LEFT JOIN incentives ON EMPLOYEE\_ID = EMPLOYEE\_REF\_ID;

1. Create View OF Employee table in which store first name, last name and salary only.

Ans: CREATE VIEW employee\_info AS SELECT first\_name, last\_name, salary FROM employee;

1. Create Procedure to find out department wise highest salary.

Ans: CREATE OR REPLACE PROCEDURE GetHighestSalaryPerDept AS BEGIN SELECT EmpID, MAX(Salary) AS HighestSalary FROM EmpDetails GROUP BY EmpID;

END;

1. Create after Insert trigger on Employee table which insert records in view table.

Ans: CREATE TRIGGER after\_insert\_employee AFTER INSERT ON Employee

FOR EACH ROW BEGIN INSERT INTO View (EmployeeID, Name, Salary, Department) VALUES (NEW.EmployeeID, NEW.Name, NEW.Salary, NEW.Department);

END;

Task -2

A. All orders for more than $1000.

Ans: SELECT \*FROM orders WHERE amount > 1000;

B. Names and cities of all salespeople in London with commission above 0.10.

Ans: SELECT name, city FROM salespeople WHERE city = 'London' AND commission > 0.10;

C. All salespeople either in Barcelona or in London

Ans: SELECT \* FROM salespeople WHERE city IN ('Barcelona', 'London');

D. All salespeople with commission between 0.10 and 0.12. (Boundary values should be excluded).

Ans: SELECT \* FROM salespeople WHERE commission > 0.10 AND commission < 0.12;

E. All customers with NULL values in city column

Ans: SELECT \* FROM customers WHERE city IS NULL;

F. All orders taken on Oct 3Rd and Oct 4th 1994.

Ans: SELECT \* FROM orders WHERE order\_date >= '1994-10-03' AND order\_date < '1994-10-05';

G. All customers serviced by peel or Motika.

Ans: SELECT \* FROM customers WHERE service\_rep IN ('Peel', 'Motika');

H. All customers whose names begin with a letter from A to B

Ans: SELECT \* FROM customers WHERE (rating > 100 OR (rating <= 100 AND city = 'Rome'));

1. All customers excluding those with rating <= 100 unless they are located in Rome.

Ans: SELECT \* FROM customers WHERE (rating > 100 OR (rating <= 100 AND city = 'Rome'));

J. All orders except those with 0 or NULL value in amt field.

Ans: SELECT \* FROM orders WHERE amount IS NOT NULL AND amount <> 0;

K. Count the number of salespeople currently listing orders in the order table.

Ans: SELECT COUNT(DISTINCT salesperson\_id) AS num\_salespeople FROM orders;