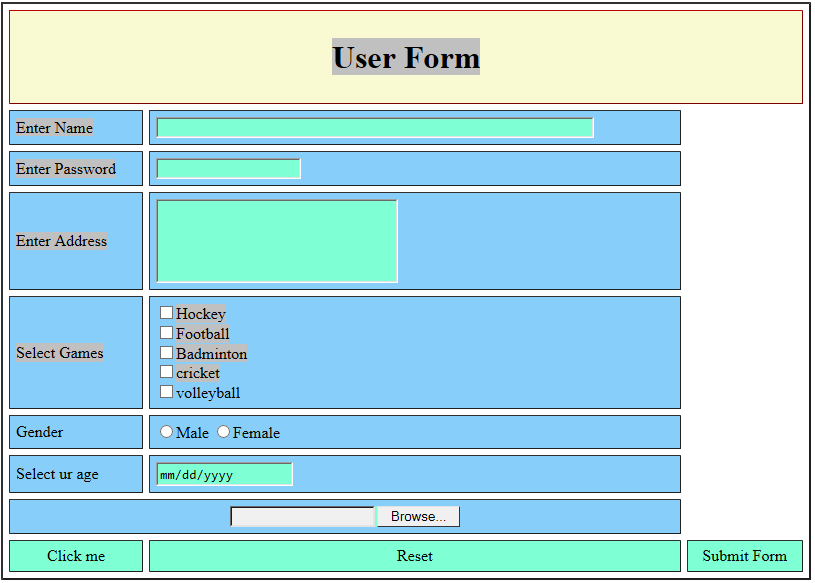
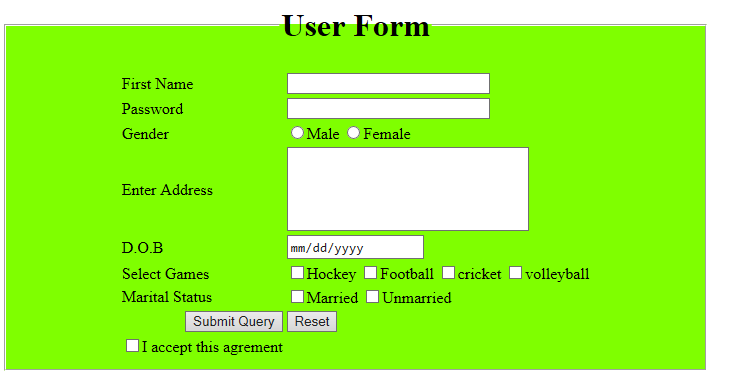
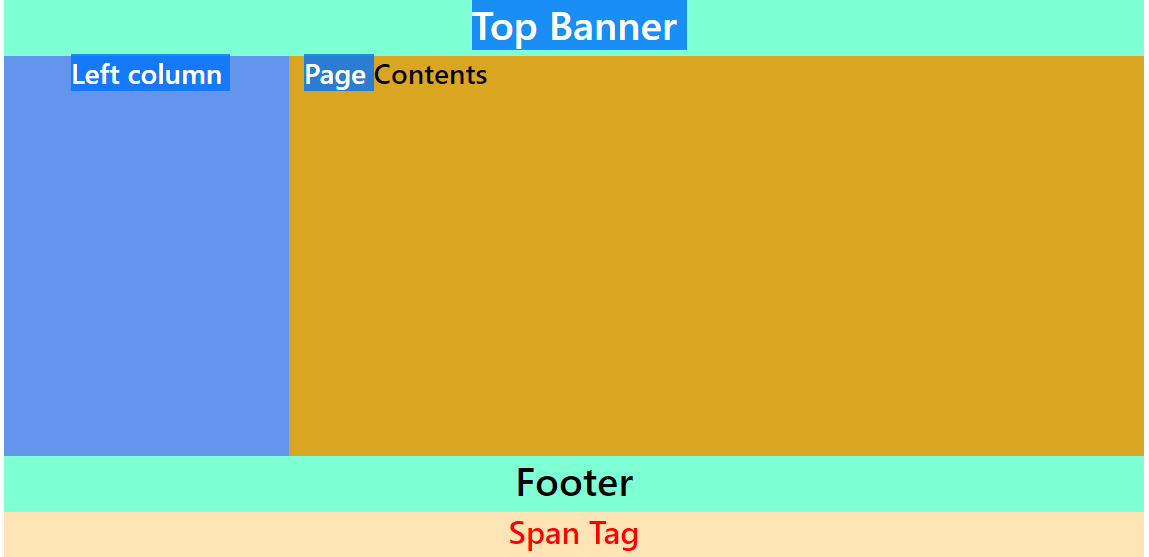
1. Try To Create the Following Form



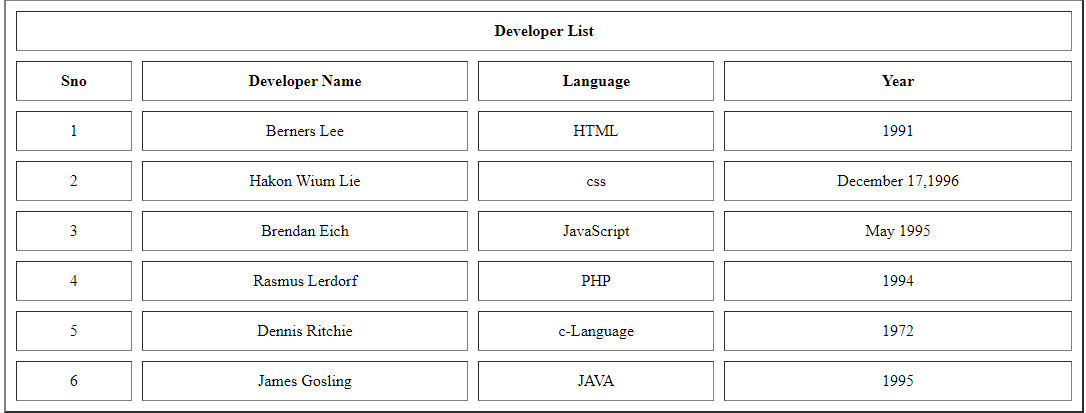
1. Try To Create Registration Form.



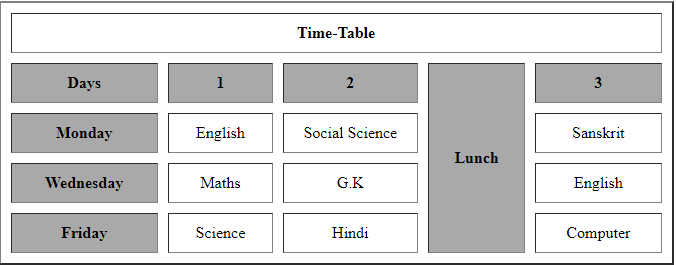
1. Using Div tag divide the page into sections.



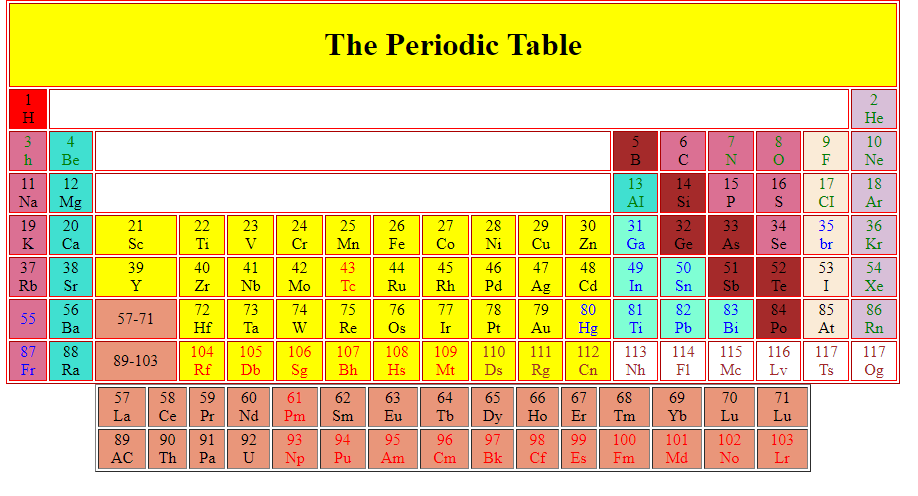
1. Table shows the list of developers:



1. Time-Table using table tag:

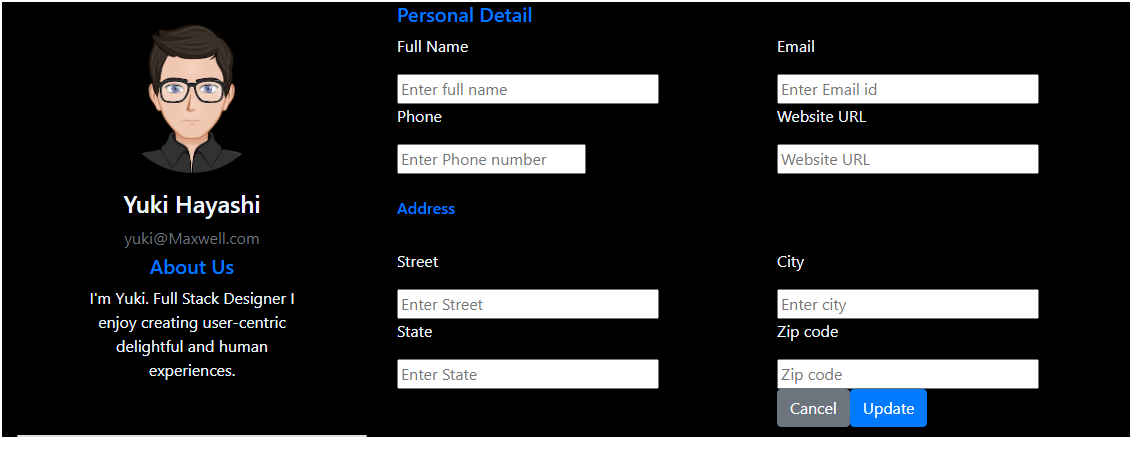


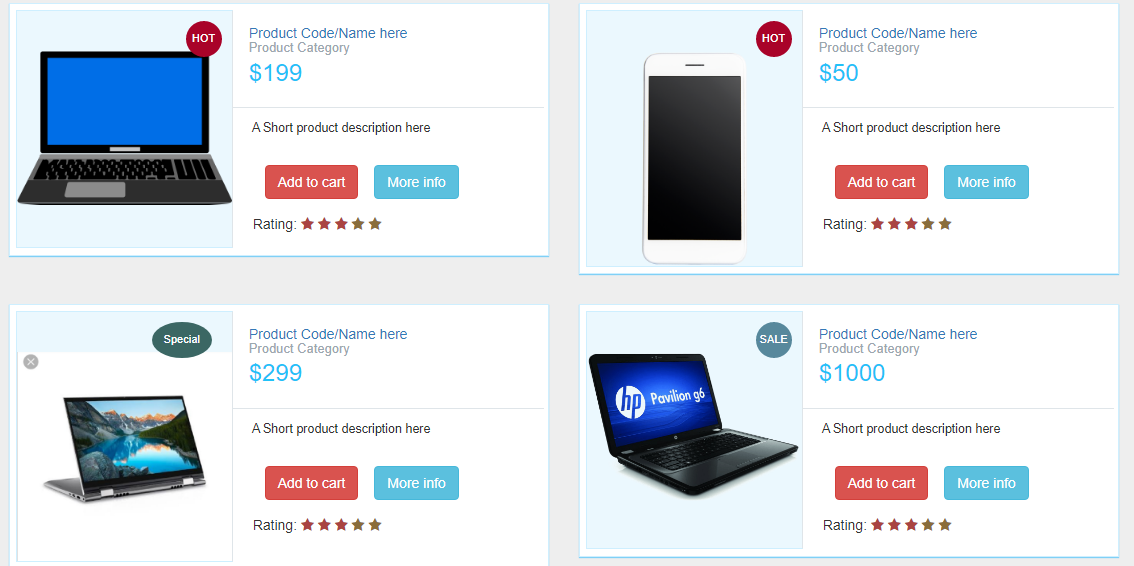
1. Periodic table using table tag:



**BOOTSRAP**

Exercise 1: Create Mange Profile Page:



Exercise 2: Create Product Listing Page:  


**Topic – SQL**

B1 Get First\_Name from employee table using alias name “Employee Name”.

SELECT First\_name"EMPLOYEE NAME" FROM employee

B2 Get FIRST\_NAME, Joining year, Joining Month and Joining Date from employee table

B2. Get FIRST\_NAME, Joining year, Joining Month and Joining Date from employee table

Select Employee\_Name, year(joining\_date) JoinYear , month(joining\_date), day(joining\_date) from EMPLOYE

B3. Get all employee details from employee table order by first\_name Ascending and Salary descending.

SELECT \*FROM employee e ORDER BY e.Employee\_name ASC, e.Salary DESC;

B4..Get employee details from the employee table whose first name contains 'o'

SELECT \* FROM employee WHERE Employee\_Name LIKE '%O%';

11..Get employee details from employee table whose joining month is "january".

SELECT \* FROM `employee` WHERE Joining\_date="2013-01-01";

12..Get department total salary with respect to a department from employee table order by total salary descending.

Select \* from employee order by Employee\_Name asc,Salary desc;

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

SELECT Department,max(Salary) FROM employee ORDER by Salary DESC .... SELECT MAX(Salary) as max\_salary from employee;

15..Select first\_name, incentive amount from employee and incentives table for those

employees who have incentives and incentive amount greater than 3000

SELECT First\_name,incentive\_Amount FROM employee a inner join incentives B on A.emloyee\_id=B.Employee\_ref\_id and incentives\_amount>3000;

16..A1. Select 2nd Highest salary from employee table.

SELECT \* FROM employee GROUP BY Salary ORDER by Salary DESC limit 1,1

17..Select first\_name, incentive amount from employee and incentives table for all

employees who got incentives using left join.

SELECT Employee\_name,nvl (incentive\_amount,0)FROM employee A left join incentives B on A.Employee\_id=B.Employee\_ref\_id

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

CREATE view employees\_view as SELECT First\_name,Last\_name,Salary FROM employee // crete view anyname as select col,col,col from table name.

A4. Create Procedure to find out department wise highest salary.

SELECT Deparment, max(Salary) as max\_salary FROM employee GROUP by Deparment

task = 2

B1. All orders for more than $1000.

SELECT \* FROM `order` WHERE AMT>1000;

B2. Names and cities of all salespeople in London with commission above 0.10.

SELECT SName,City from `salaperson` WHERE COMM > 10 AND City ='Londan';

B3. All salespeople either in Barcelona or in London.

SELECT SName,City FROM `salaperson` WHERE City in ('barcelone','Lonadan');

B4. All salespeople with commission between 0.10 and 0.12. (Boundary values should

be excluded).

SELECT COMM , SName FROM `salaperson` WHERE COMM > 10 AND COMM < 12;

B5. All customers with NULL values in city column.

SELECT CNAME from `customer` WHERE CITY is null;

I1. All orders taken on Oct 3Rd and Oct 4th 1994.

Select \* from `order`where ODE in (03-10-94,04-10-94);

I2. All customers serviced by peel or Motika.

Select CName from customer, `order` where `order`.SName = customer.SNO and `order`.SName in( select SNO

from salaperson where SName in 'Peel','Motika';

I3. All customers whose names begin with a letter from A to B

SELECT CNAME FROM `customer` WHERE CNAME like 'A%'or CNAME like 'B%';

I4. All customers excluding those with rating <= 100 unless they are located in Rome.

SELECT CName from `customer` WHERE CITY = 'Rome' AND RATING <=100;

A1. All orders except those with 0 or NULL value in amt field.

SELECT CNM. `order` WHERE AMT != 0 or AMT is NOT null;

select count(distinct SNO) from `order`;

A3. Largest order taken by each salesperson, datewise.

Select ODE,SNO, max(Amt) from `order` group by ODE,SNO order by ODE ,SNO;

A4. Largest order taken by each salesperson with order value more than

$3000. (ie same city).

Select ODE, SNO, max(Amt) from `order` where Amt > 3000 group by ODE,SNO order by ODE,SNO;

SELECT COMM , SName FROM `salaperson` WHERE COMM > 10 AND COMM < 12;

B5. All customers with NULL values in city column.

SELECT CNAME from `customer` WHERE CITY is null;

I1. All orders taken on Oct 3Rd and Oct 4th 1994.

Select \* from `order`where ODE in (03-10-94,04-10-94);

I2. All customers serviced by peel or Motika.

Select CName from customer, `order` where `order`.SName = customer.SNO and `order`.SName in( select SNO

from salaperson where SName in 'Peel','Motika';

I3. All customers whose names begin with a letter from A to B

SELECT CNAME FROM `customer` WHERE CNAME like 'A%'or CNAME like 'B%';

I4. All customers excluding those with rating <= 100 unless they are located in Rome.

SELECT CName from `customer` WHERE CITY = 'Rome' AND RATING <=100;

A1. All orders except those with 0 or NULL value in amt field.

SELECT CNM. `order` WHERE AMT != 0 or AMT is NOT null;

select count(distinct SNO) from `order`;

A3. Largest order taken by each salesperson, datewise.

Select ODE,SNO, max(Amt) from `order` group by ODE,SNO order by ODE ,SNO;

A4. Largest order taken by each salesperson with order value more than

$3000. (ie same city).

Select ODE, SNO, max(Amt) from `order` where Amt > 3000 group by ODE,SNO order by ODE,SNO;

TASK :3

B1.Return the FirstName, LastName, ProductName, and SalePrice for all

products sold in the month of October2005

SELECT FirstName, LastName, ProductName, SalesPrice FROM customers INNER JOIN product ON product.ProductID=Customers.CustomerID INNER JOIN Sales ON SalesID = CustmerID WHERE SalesDate BETWEEN '2005-10-01' AND '2005-10-31';

B2. Return the CustomerID, FirstName, and LastName of those individuals in

the Customer table who have made no Sales purchases.

SELECT CustomerID,FirstName,LastName FROM customers WHERE CustomerID not in (select distinct CustomerID from Sales);

3. Return the FirstName, LastName, SalePrice, Recommended SalePrice, and

the difference between the SalePrice and Recommended SalePrice for all

Sales. The difference must be returned as a positive number.

SELECT FirstName,LastName,SalesPrice as diff FROM customers INNER JOIN product ON product.ProductID = Customers.CustomerID INNER JOIN sales ON SalesID = SalesID;

B4. Return the average SalePrice by Product Category.

SELECT Category,AVG (salesprice ) as total FROM product inner join sales on product = productID GROUP BY Category

CREATE PROCEDURE spInsertInToSalesDetails()

DECLARE @firstName varchar(50);

@lastName varchar(50)

@productName varchar(50);

@cat varchar(50);

@sPrice float;

@custID int;

@pid int;

B5.Return the Product Category and the average Sale Price for those customers

who have purchased two or more products.

SELECT distinct FirstName,LastName,Category, COUNT(\*) AS 'Total Product',AVG(salesprice) as 'Average Sale Price' FROM product INNER JOIN sales ON product.ProductID= sales.ProductID INNER JOIN customers ON sales.CustmerID=customers.CustomerID GROUP BY category,FirstName,LastName HAVING COUNT(\*)>=2;

Update the Sale Price to the Recommended Sale Price of those Sales

occurring between 6/10/2005and6/20/2005.

UPDATE `sales` SET `SalesPrice`='RecommendedPr' FROM sales INNER JOIN products ON sales.ProductID = product.ProductID WHERE SalesDate BETWEEN '2005-06-10' AND '2005-06-20';

A2. Number of Sales by Product Category where the average Recommended

Priceis10 or more dollars greater than the average Sale Price.

task 4

B1. Write an SQL query that returns the project number and name for projects

with a budget greater than $100,000.

SELECT \* FROM `proj` where budget>50000;

B2. Write an SQL query that returns all works on records where hours worked

is less than10and the responsibility is “Manager”.

SELECT \* FROM workson WHERE hours < 10 AND resp = 'Manager';

B3. Write an SQL query that returns the employees (number and name only)

who have a title of “EEE‟ or “SA‟ and make more than $35,000.

SELECT eno, ename FROM emp WHERE (title = 'EE' OR title = 'SA') AND salary > 35000;

B4. Write an SQL query that returns the employees (name only) in department

“D1‟ordered by decreasing salary

SELECT ename FROM emp WHERE dno = 'D1' ORDER BY salary DESC;

4) Write an SQL query that returns the employees (name only) in department 'D1'

ordered by decreasing salary

SELECT \* FROM dept ORDER BY dname ASC;

5) Write an SQL query that returns the departments (all fields) ordered by ascending

department name

SELECT ename, dname, title

FROM emp, dept

WHERE emp.dno = dept.dno;

6) Write an SQL query that returns the employee name, department name, and employee

title.

SELECT pname, hours, proj.pno

FROM workson, proj

WHERE hours > 10 AND proj.pno = workson.pno;

7) Write an SQL query that returns the project name, hours worked, and project number

for all works on records where hours > 10.

SELECT pname, hours, proj.pno FROM workson, proj WHERE hours > 10 AND proj.pno = workson.pno;

8) Write an SQL query that returns the project name, department name, and budget for

all projects with a budget < $50,000.

SELECT pname, dname, budget

FROM proj, dept

WHERE budget < 50000 AND proj.dno = dept.dno;

9) Write an SQL query that returns the employee numbers and salaries of all employees

in the 'Consulting' department ordered by descending salary.

SELECT eno, salary

FROM emp, dept

WHERE dname = 'Consulting'

ORDER BY salary DESC

10) Write an SQL query that returns the employee name, project name, employee title,

and hours for all works on records

SELECT ename, pname, title

FROM emp, proj, workson

WHERE emp.eno = workson.eno and proj.pno = workson.pno;