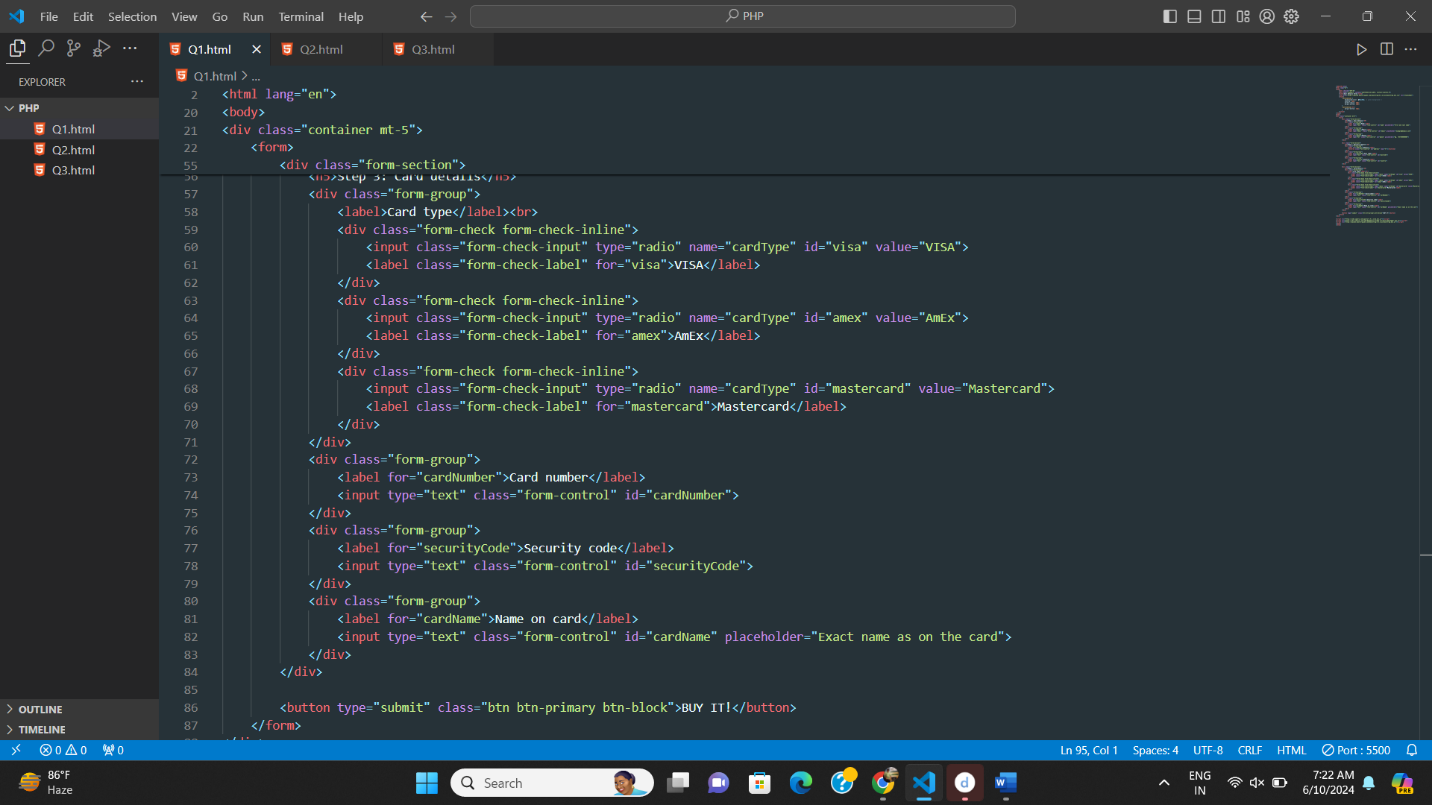
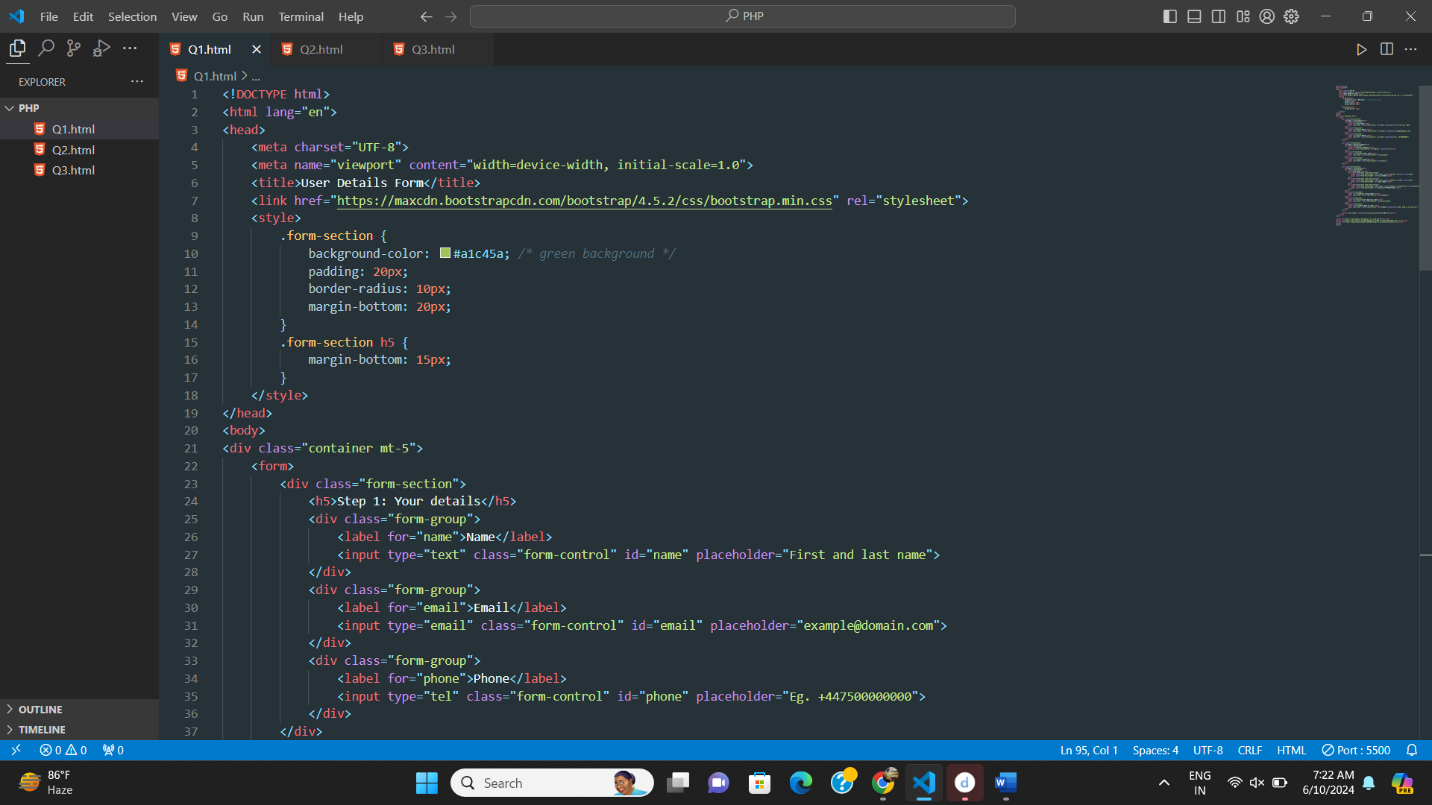
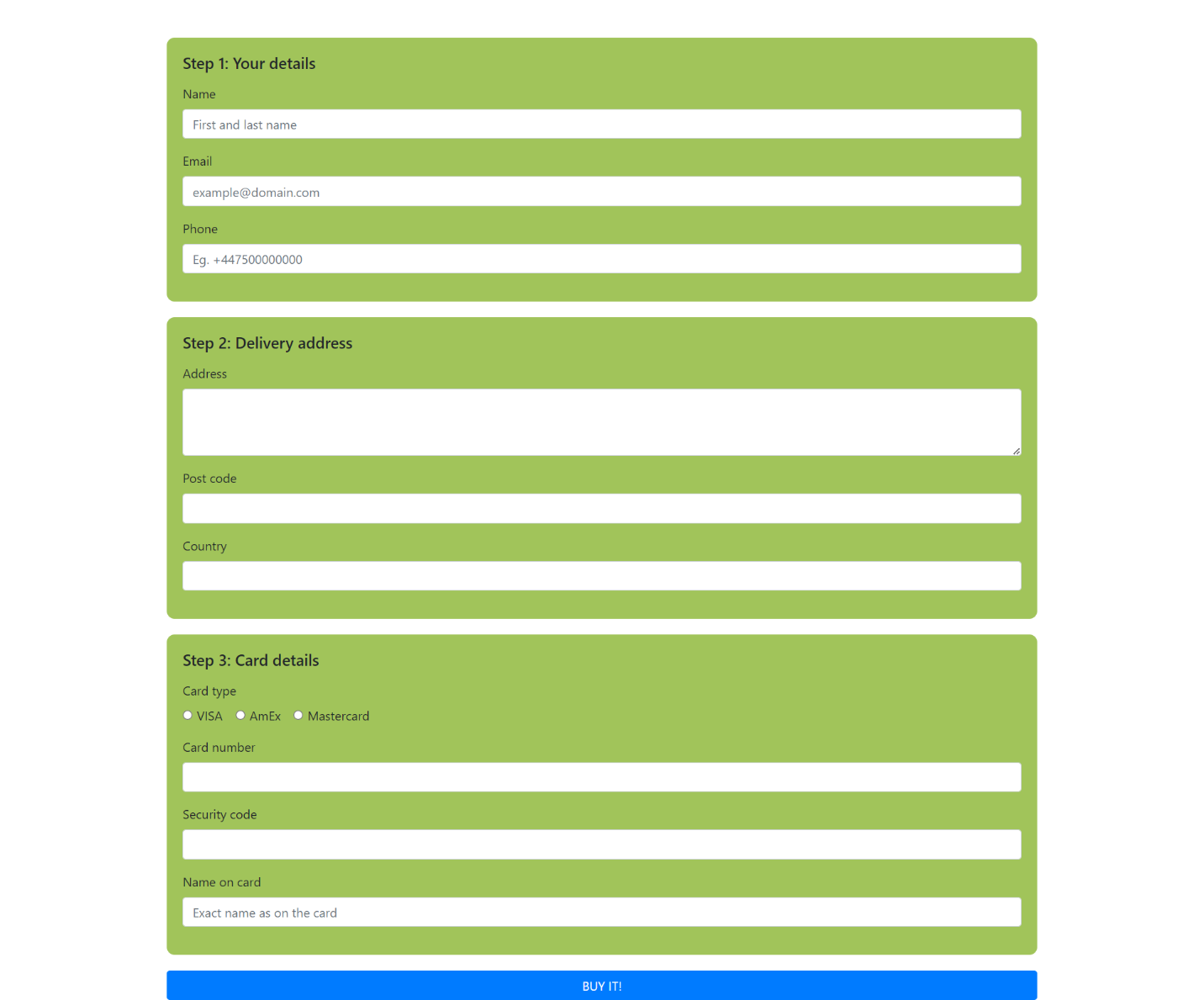
**PHP Assignment**

**MODULE – 2 HTML, CSS, SQL**

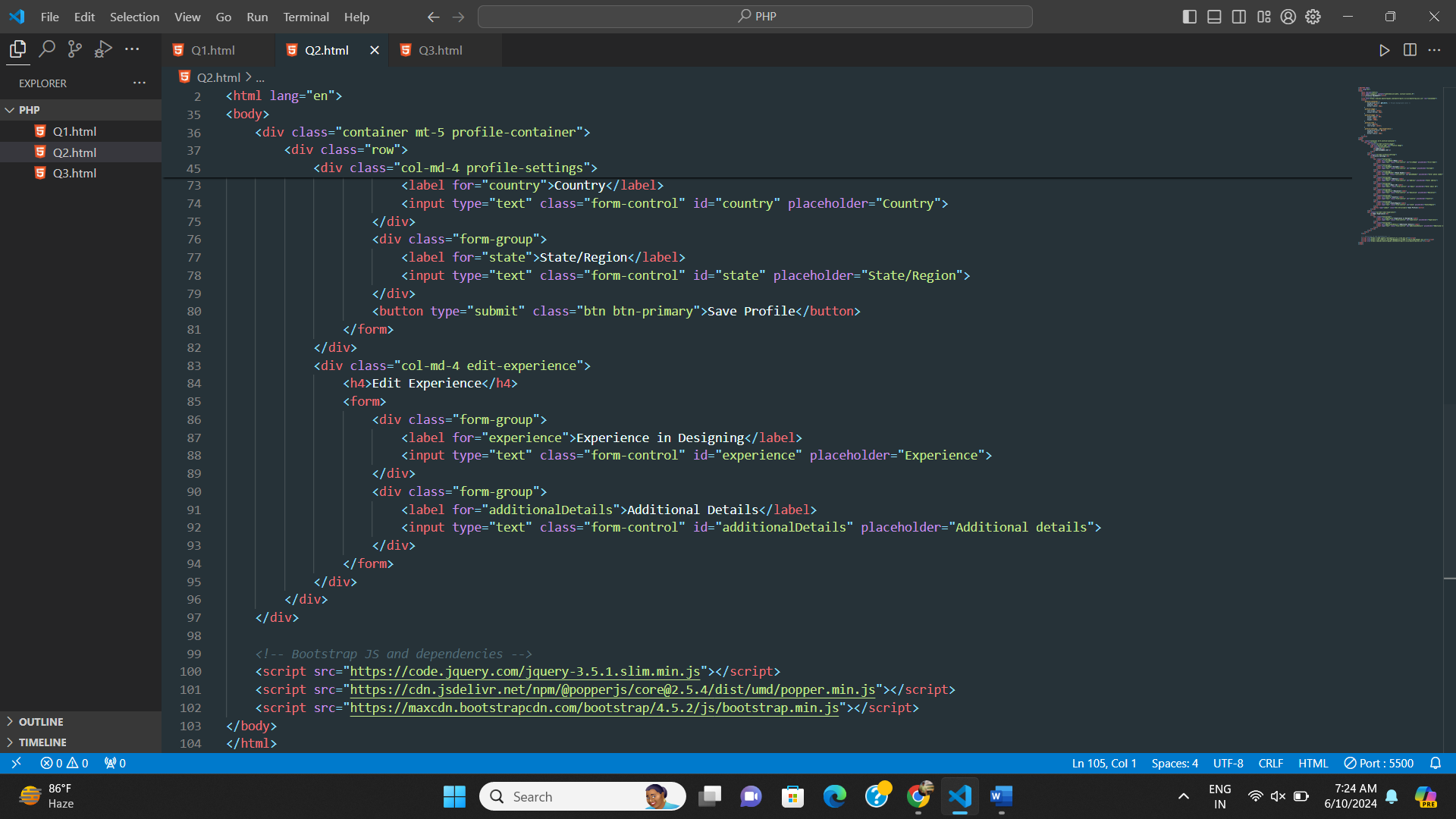
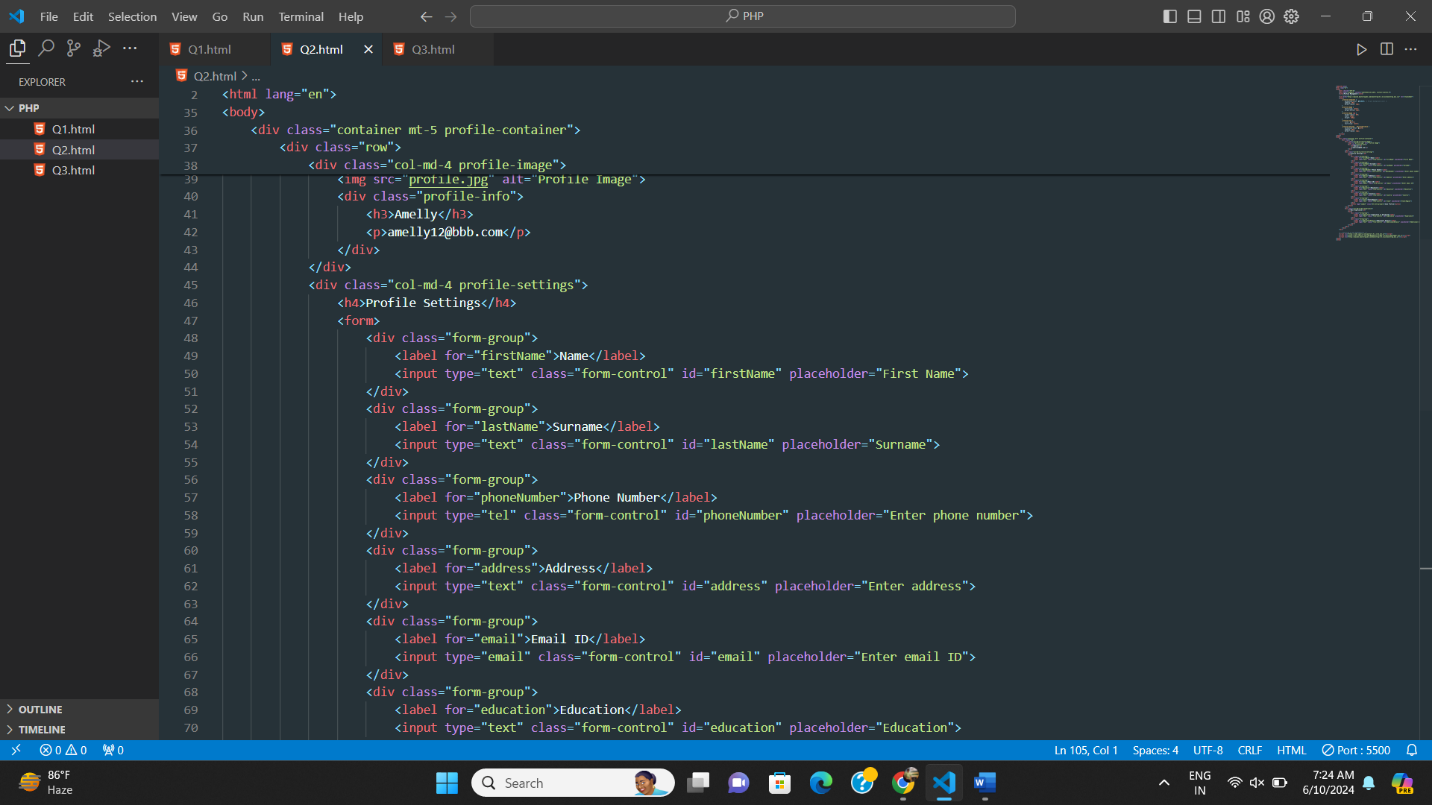
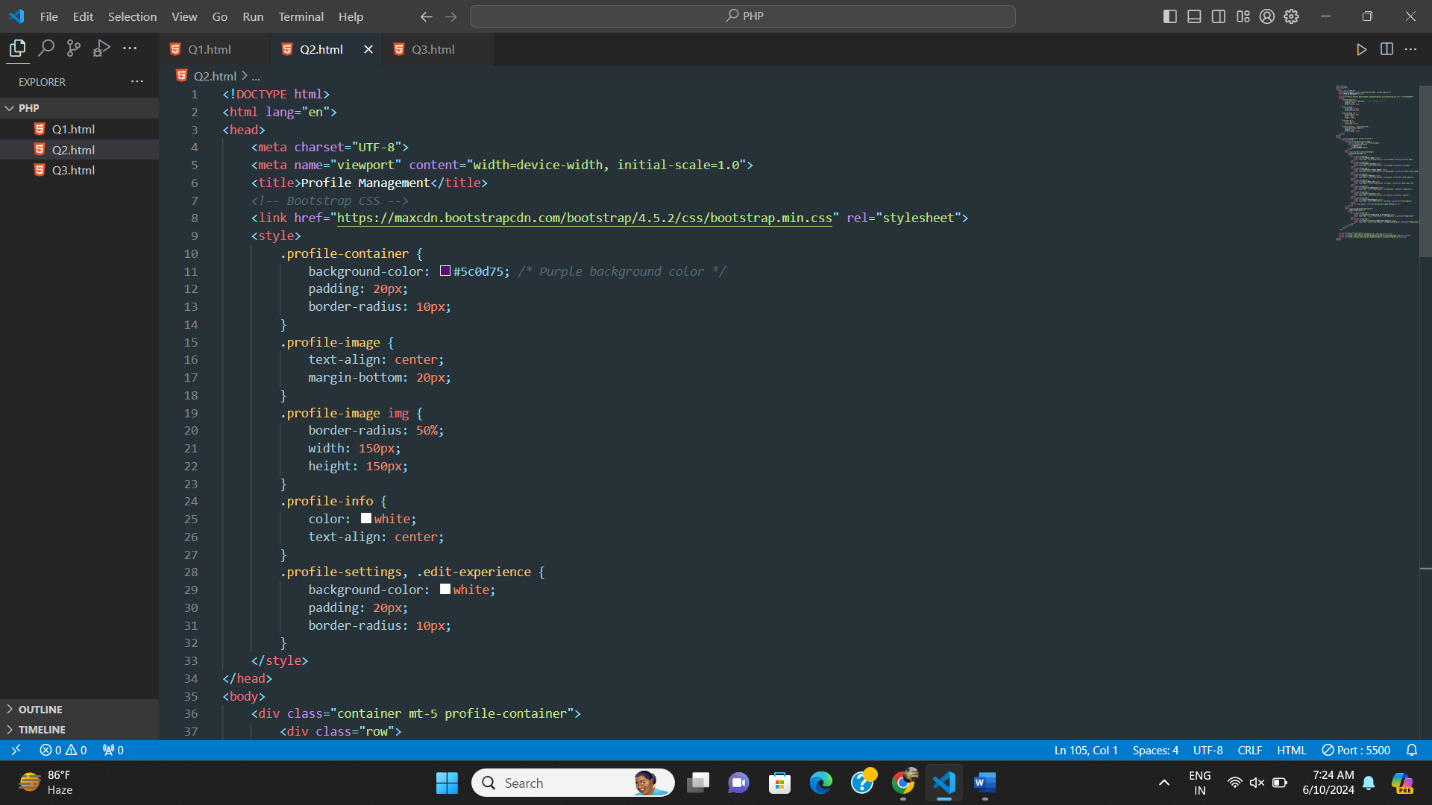
Exercise 1: Try To Create User Details Form.

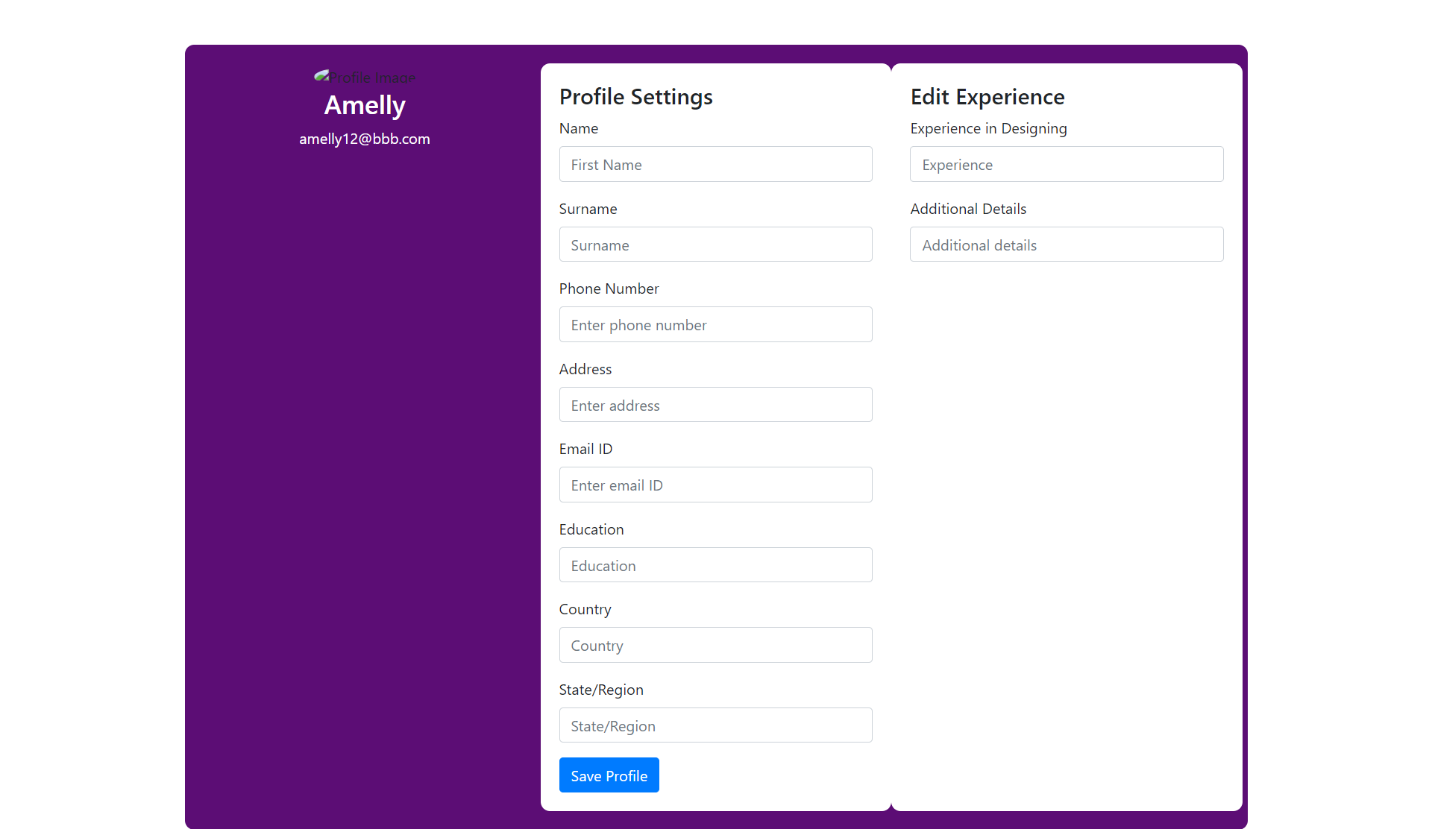




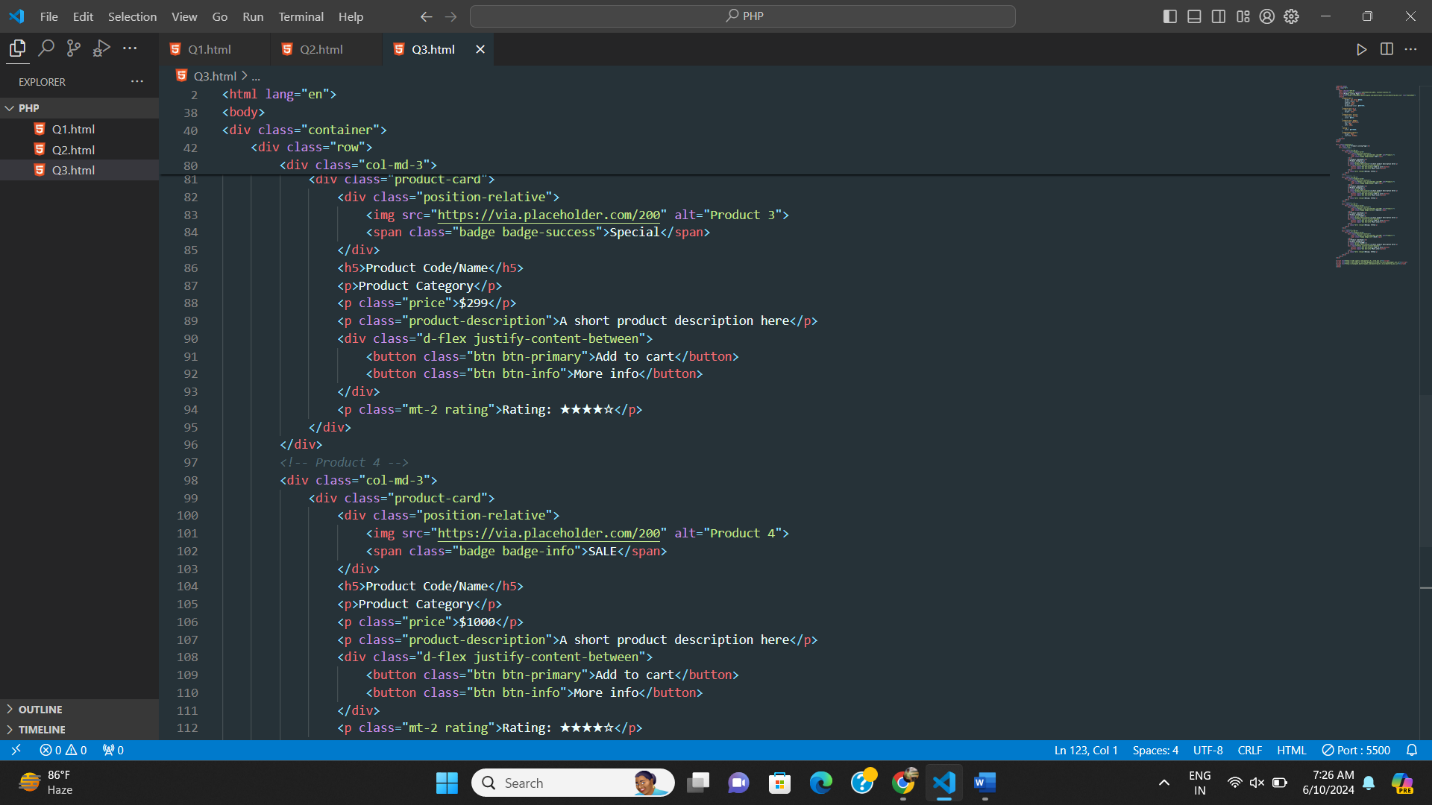
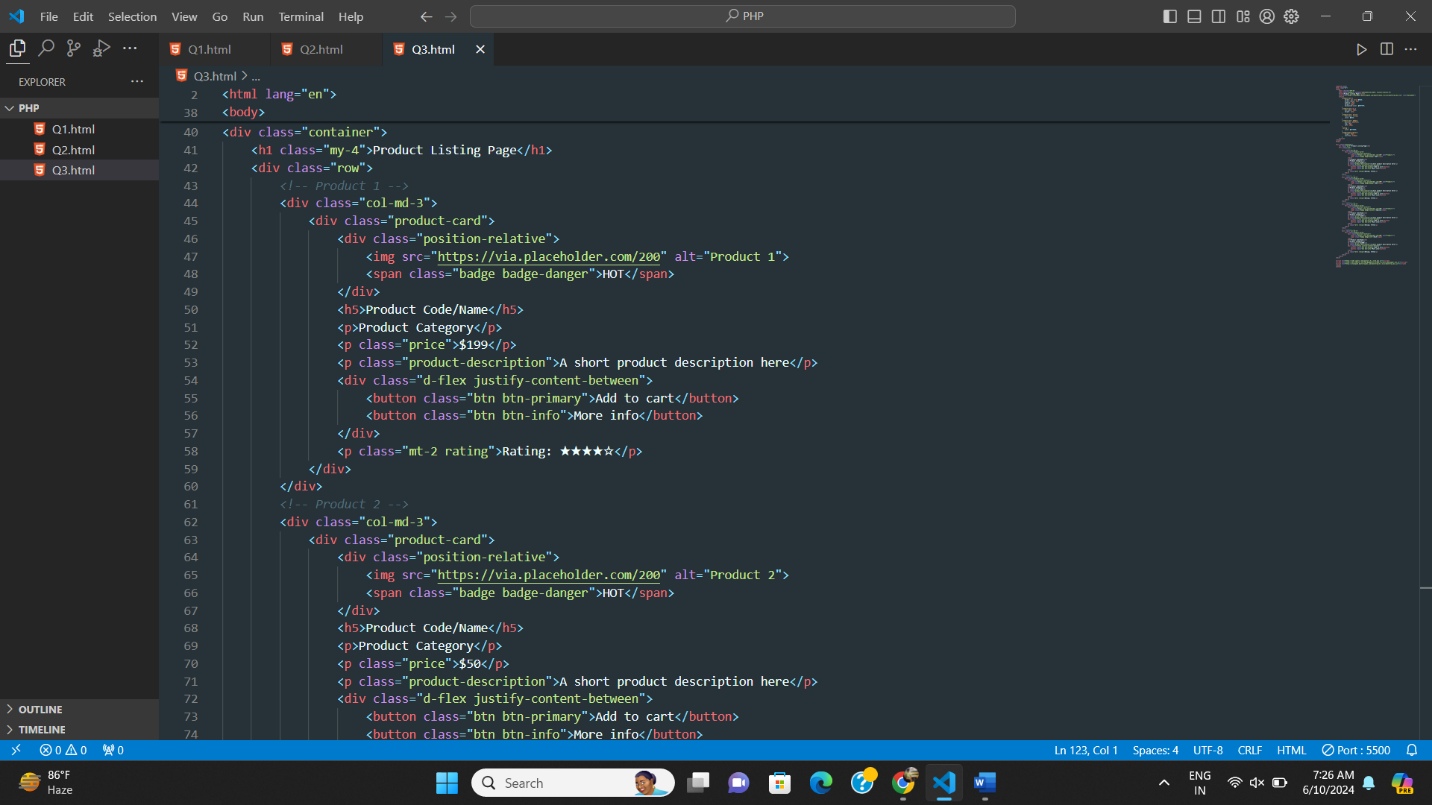
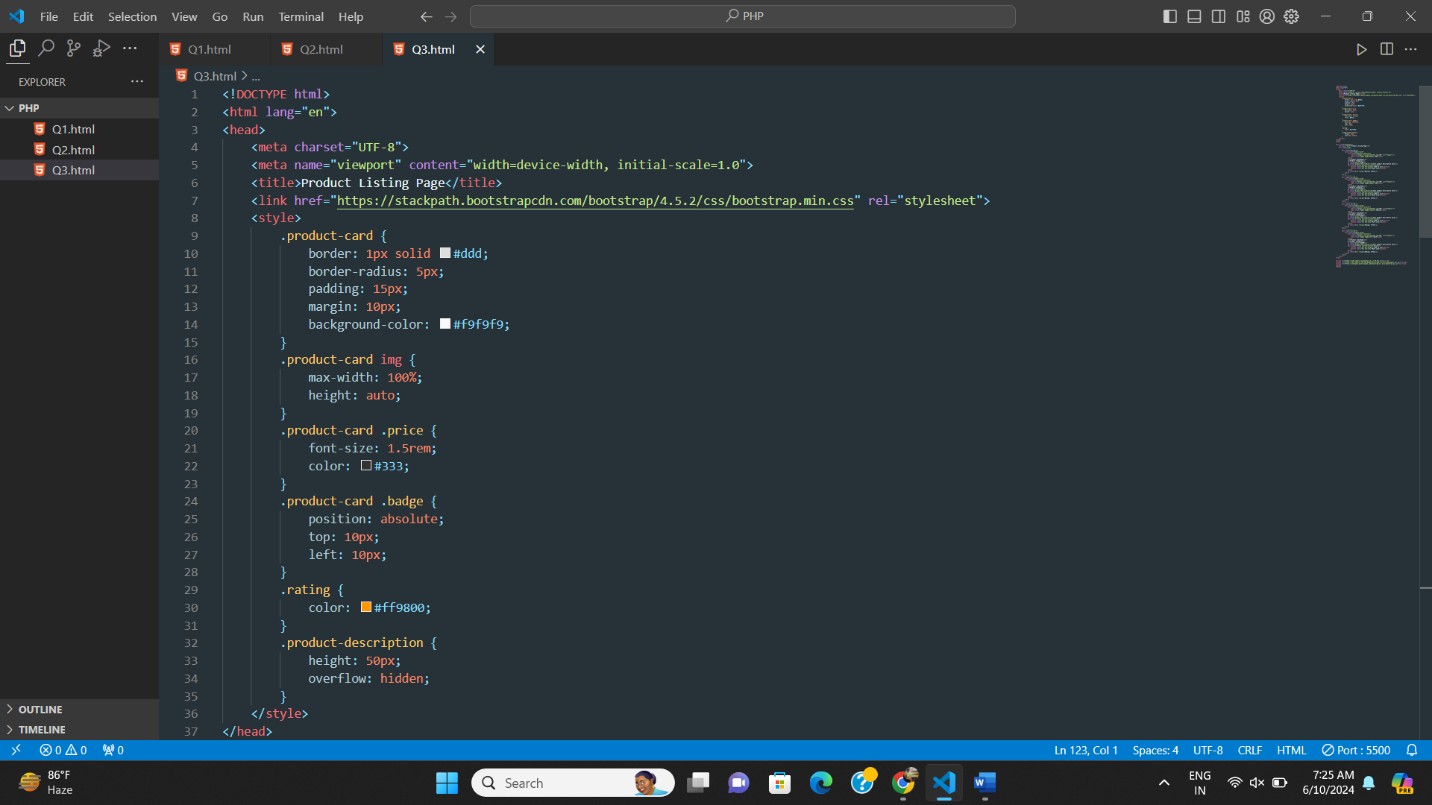
BOOTSRAP

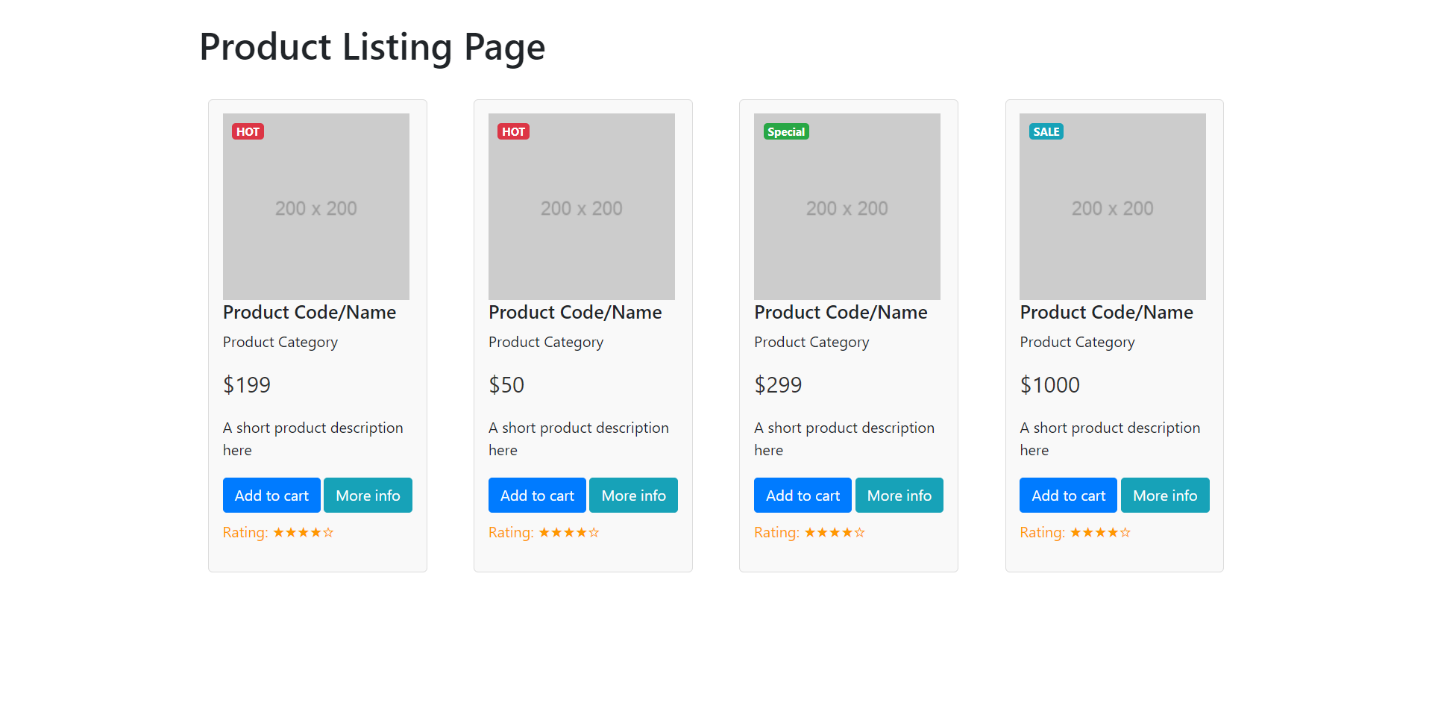
Exercise 1: Create manage Profile:





Exercise2 : CreateProductListing Page:





**--SQL Task-1**

**--a) Get First\_Name from employee table using alias name “Employee Name”.**

SELECT first\_name AS "Employee Name" FROM Employee;

**--b) Get FIRST\_NAME, Joining year, Joining Month and Joining Date from employee table.**

SELECT

first\_name,

YEAR(joining\_Date) AS joining\_Year,

MONTH(joining\_Date) AS joining\_Month,

DAY(joining\_Date) AS joining\_Date

FROM Employee;

**--c) Get all employee details from the employee table order by First Name Ascending And Salary descending?**

-- First Name Ascending

SELECT \* FROM Employee

ORDER BY first\_name ASC;

--Salary Descending

SELECT \* FROM Employee

ORDER BY salary DESC;

**--d) Get employee details from employee table whose first name contains „o‟.**

SELECT \* FROM Employee

WHERE first\_name LIKE 'o%';

**--e) Get employee details from employee table whose joining month is “January”.**

SELECT \* FROM Employee

WHERE MONTH(joining\_date) = 1;

**--f) Get department, total salary with respect to a department from employee table**

--Order By total salary descending.

SELECT department, SUM(salary) AS total\_salary

FROM Employee

GROUP BY department

ORDER BY total\_salary DESC;

**-- g) Get department-wise maximum salary from employee table ordered by salary ascending**

SELECT department, MAX(salary) AS max\_salary

FROM Employee

GROUP BY department

ORDER BY max\_salary ASC;

**-- h) Select first\_name, incentive amount from employee and incentives table for those employees who have incentives and incentive amount greater than 3000**

SELECT e.first\_name, i.incentive\_amount

FROM Employee e

JOIN Incentive i ON e.em\_id = i.employee\_ref\_id

WHERE i.incentive\_amount > 3000;

**-- i) Select 2nd highest salary from employee table**

SELECT MAX(salary) AS second\_highest\_salary

FROM Employee

WHERE salary < (SELECT MAX(salary) FROM Employee);

**-- j) Select first\_name, incentive amount from employee and incentives table for all employees who got incentives using LEFT JOIN**

SELECT e.first\_name, i.incentive\_amount

FROM Employee e

LEFT JOIN Incentive i ON e.em\_id = i.employee\_ref\_id;

**-- k) Create a view of Employee table which stores first name, last name, and salary only**

CREATE VIEW EmployeeView AS

SELECT first\_name, last\_name, salary

FROM Employee;

**-- l) Create a procedure to find out department-wise highest salary**

CREATE PROCEDURE GetDepartmentWiseHighestSalary

AS

BEGIN

SELECT department, MAX(salary) AS highest\_salary

FROM Employee

GROUP BY department;

END;

**-- m) Create an AFTER INSERT trigger on Employee table which inserts records into the view table**

-- Creating the view table

CREATE TABLE EmployeeViewTable (

first\_name VARCHAR(100),

last\_name VARCHAR(100),

salary BIGINT

);

-- Creating the trigger

CREATE TRIGGER after\_employee\_insert

ON Employee

AFTER INSERT

AS

BEGIN

INSERT INTO EmployeeViewTable (first\_name, last\_name, salary)

SELECT inserted.first\_name, inserted.last\_name, inserted.salary

FROM inserted;

END;

**--SQL TASK-2**

**--a) All orders for more than $1000.**

SELECT \*

FROM Orders

WHERE amt > 1000;

**--b) Names and cities of all salespeople in London with commission above 0.10.**

SELECT sname, city

FROM Salesperson

WHERE city = 'London' AND comm > 0.10;

**--c) All salespeople either in Barcelona or in London.**

SELECT \*

FROM Salesperson

WHERE city IN ('Barcelona', 'London');

**--d) All salespeople with commission between 0.10 and 0.12. (Boundary values should**

**--be excluded).**

SELECT \*

FROM Salesperson

WHERE comm > 0.10 AND comm < 0.12;

**--e) All customers with NULL values in city column.**

SELECT \*

FROM Customer

WHERE city IS NULL;

**--f) All orders taken on Oct 3Rd and Oct 4th 1994.**

SELECT \*

FROM Orders

WHERE ode BETWEEN '1994-10-03' AND '1994-10-04';

**--g) All customers serviced by peel or Motika.**

SELECT \*

FROM Customer

WHERE sno IN (

SELECT sno FROM Salesperson WHERE sname IN ('Peel', 'Motika')

);

**--h) All customers whose names begin with a letter from A to B**

SELECT \*

FROM Customer

WHERE cname LIKE 'A%' OR cname LIKE 'B%';

**--i) All customers excluding those with rating <= 100 unless they are located in Rome.**

SELECT \*

FROM Customer

WHERE (rating <= 100 OR city = 'Rome') AND rating IS NOT NULL;

**--j) All orders except those with 0 or NULL value in amt field**

SELECT \*

FROM Orders

WHERE amt IS NOT NULL AND amt != 0;

**--k) Count the number of salespeople currently listing orders in the order table.**

SELECT COUNT(DISTINCT sno) AS num\_salespeople

FROM Orders;