

MERN Stack Assignment Set

Module 1 – SE - Overview of IT Industry

1) What is a Program?

LAB EXERCISE: Write a simple "Hello World" program in two different programming languages of your choice. Compare the structure and syntax.

Ans :

hello.c

```
#include <stdio.h>
```

```
int main() {
```

```
    // Print message to the console
```

```
    printf("Hello World\n");
```

```
    return 0;
```

```
}
```

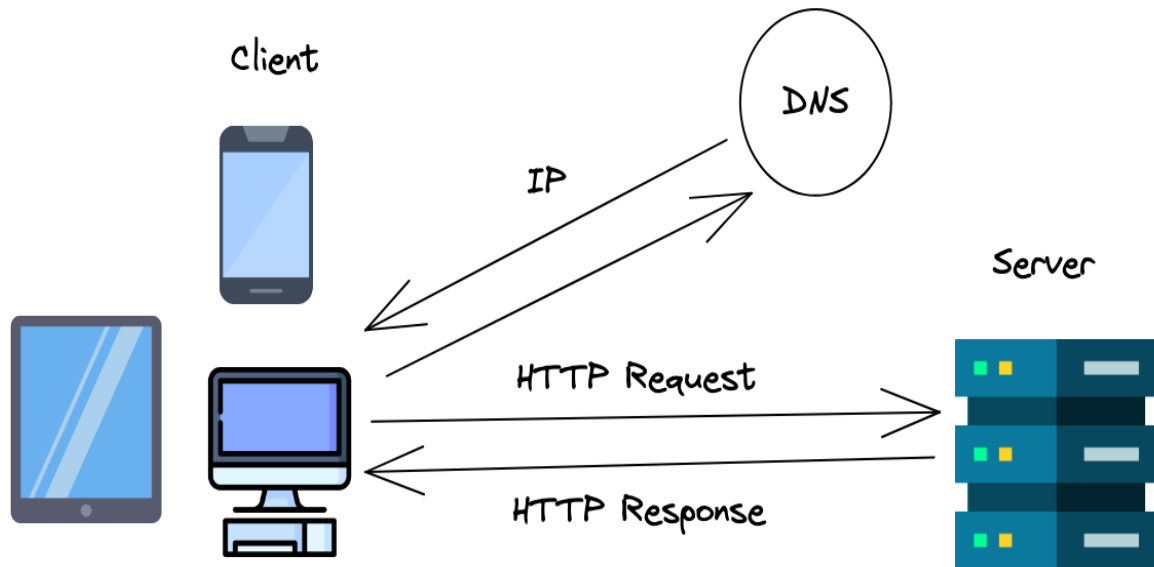
hello.js

```
console.log("Hello World");
```

2) World Wide Web & How Internet Works

LAB EXERCISE: Research and create a diagram of how data is transmitted from a client to a server over the internet.

Ans :



3) Network Layers on Client and Server

LAB EXERCISE: Design a simple HTTP client-server communication in any language.

Ans :

Server (server.py):

```
from http.server import BaseHTTPRequestHandler, HTTPServer
```

```
class MyServer(BaseHTTPRequestHandler):
```

```
    def do_GET(self):
```

```
        self.send_response(200)
```

```
        self.end_headers()
```

```
        self.wfile.write(b"Hello from Server!")
```

```
server = HTTPServer(('localhost', 8080), MyServer)
```

```
print("Server running on http://localhost:8080")
```

```
server.serve_forever()
```

Client (client.py):

```
import http.client

conn = http.client.HTTPConnection("localhost", 8080)
conn.request("GET", "/")
res = conn.getresponse()
print(res.read().decode())
conn.close()
```

4)Types of Internet Connections

LAB EXERCISE: Research different types of internet connections (e.g., broadband, fiber, satellite) and list their pros and cons.

Ans :

Broadband (DSL/Cable)

Pros: Fast, always on, widely available

Cons: Speed may reduce during peak hours

Fiber Optic

Pros: Very high speed, reliable, good for streaming/gaming

Cons: Costly, not available everywhere

Satellite

Pros: Available in remote areas

Cons: Slower, expensive, weather can affect signal

Mobile Data (4G/5G)

Pros: Wireless, portable, fast (5G)

Cons: Limited data plans, signal issues

Dial-up

Pros: Cheap, basic internet access

Cons: Very slow, blocks phone line

5) Protocols

LAB EXERCISE: Simulate HTTP and FTP requests using command line tools (e.g., curl).

Ans :

Simulate HTTP:

curl <http://example.com>

Simulate HTTPS:

curl <https://example.com>

Simulate FTP:

curl <ftp://ftp.example.com/file.txt>

- *curl is used to send requests and get data from servers.*

6) Application Security

LAB EXERCISE: Identify and explain three common application security vulnerabilities. Suggest possible solutions.

Ans:

SQL Injection

What: Fake login using code like ' OR '1'='1

Fix: Use safe queries (prepared statements)

XSS (Cross-Site Scripting)

What: Runs script like <script>alert('Hi')</script>

Fix: Clean input using htmlspecialchars()

Broken Authentication

What: Weak passwords or no logout

Fix: Use strong password, OTP, and auto logout

LAB EXERCISE: Identify and classify 5 applications you use daily as either system software or application software.

Ans :

Application Software = Used by user

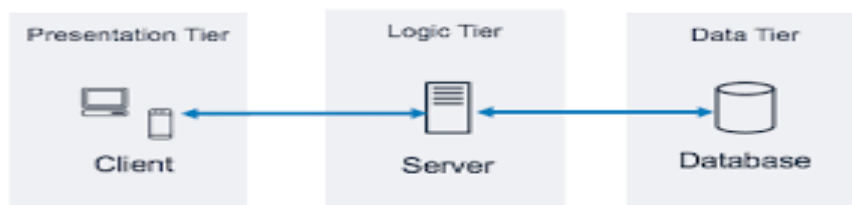
System Software = Runs the system

App	Type
Chrome	Application Software
WhatsApp	Application Software
MS Word	Application Software
Windows OS	System Software
Antivirus	System Software

7) Software Architecture

LAB EXERCISE: Design a basic three-tier software architecture diagram for a web application.

Ans:



8) Layers in Software Architecture

LAB EXERCISE: Create a case study on the functionality of the presentation, business logic, and data access layers of a given software system.

Ans:

Presentation Layer	Shows product list, cart, buttons (UI for user)
Business Logic Layer	Handles discounts, order total, payment rules
Data Access Layer	Connects to database, gets/saves product and user info

9) Software Environments

LAB EXERCISE: Explore different types of software environments (development, testing, production). Set up a basic environment in a virtual machine.

Ans:

Explore Software Environments:

1. Development Environment:

Used by developers to write and test code.

Tools: IDE (e.g., VS Code), compilers, debuggers.

2. Testing Environment:

Used to test software for bugs or issues.

Simulates real user conditions.

3. Production Environment:

Live environment where real users use the software.

Must be stable and secure.

Setup Basic Environment in a Virtual Machine:

Steps:

1. Install Virtual Machine:

Use VirtualBox or VMware.

2. Install OS in VM:

Example: Ubuntu Linux.

3. Set up Development Tools:

Install **VS** Code, Python/Java, and Git.

4. Create Folders:

/dev – for development

/test – for testing

/prod – for production simulation

5. Test with Sample Code:

Write a simple program and run it in each folder.

10) Source Code

LAB EXERCISE: Write and upload your first source code file to GitHub.

And :

GitHub Link

https://github.com/Dhruvi79/Soft-Skills/blob/main/softskills_assignment_Email.pdf

11) GitHub and Introductions

LAB EXERCISE: Create a GitHub repository and document how to commit and push code changes.

Ans:

1. Create Repository

Go to [GitHub.com](https://github.com)

Click “+” → New repository

Name it (e.g., my-project)

Click Create repository

2. Upload Code

Inside the repo, click “Add file” → “Upload files”

Choose your source code file (e.g., hello.py)

Click “Commit changes” at the bottom

Done!

12) Student Account in GitHub

LAB EXERCISE: Create a student account on GitHub and collaborate on a small project with a classmate.

Ans:

Create Account

Sign up at github.com using your email.

Create Repository

Click “+” → New repository

Name it (e.g., student-project) → Create

Add Collaborator

Go to Settings → Collaborators

Add classmate’s username → Invite

Collaborate

Both upload or edit code in the repo.

13) Types of Software

LAB EXERCISE: Create a list of software you use regularly and classify them into the following categories: system, application, and utility software.

Ans:

System Software

Windows 10

Android

Application Software

MS Word

Google Chrome

WhatsApp

Utility Software

Windows Defender

WinRAR

CCleaner

14) GIT and GITHUB Training

LAB EXERCISE: Follow a GIT tutorial to practice cloning, branching, and merging repositories.

Ans :

Clone a Repository

Go to GitHub

Click “Code” → Download ZIP

Extract the ZIP file on your computer

Create a Branch

Use GitHub Desktop → Open the repo

Click “Current Branch” → New Branch

Name your branch and create it

Make Changes & Merge

Edit files in your branch

Click “Commit to branch” in GitHub Desktop

Switch to main branch

Click “Merge into main”

15) Application Software

LAB EXERCISE: Write a report on the various types of application software and how they improve productivity.

Ans :

Types:

1. Word Processors – MS Word
2. Spreadsheets – MS Excel
3. Presentation Software – PowerPoint
4. Browsers – Google Chrome
5. Email Clients – Outlook
6. Media Players – VLC

Improves Productivity By:

Saving time

Reducing errors

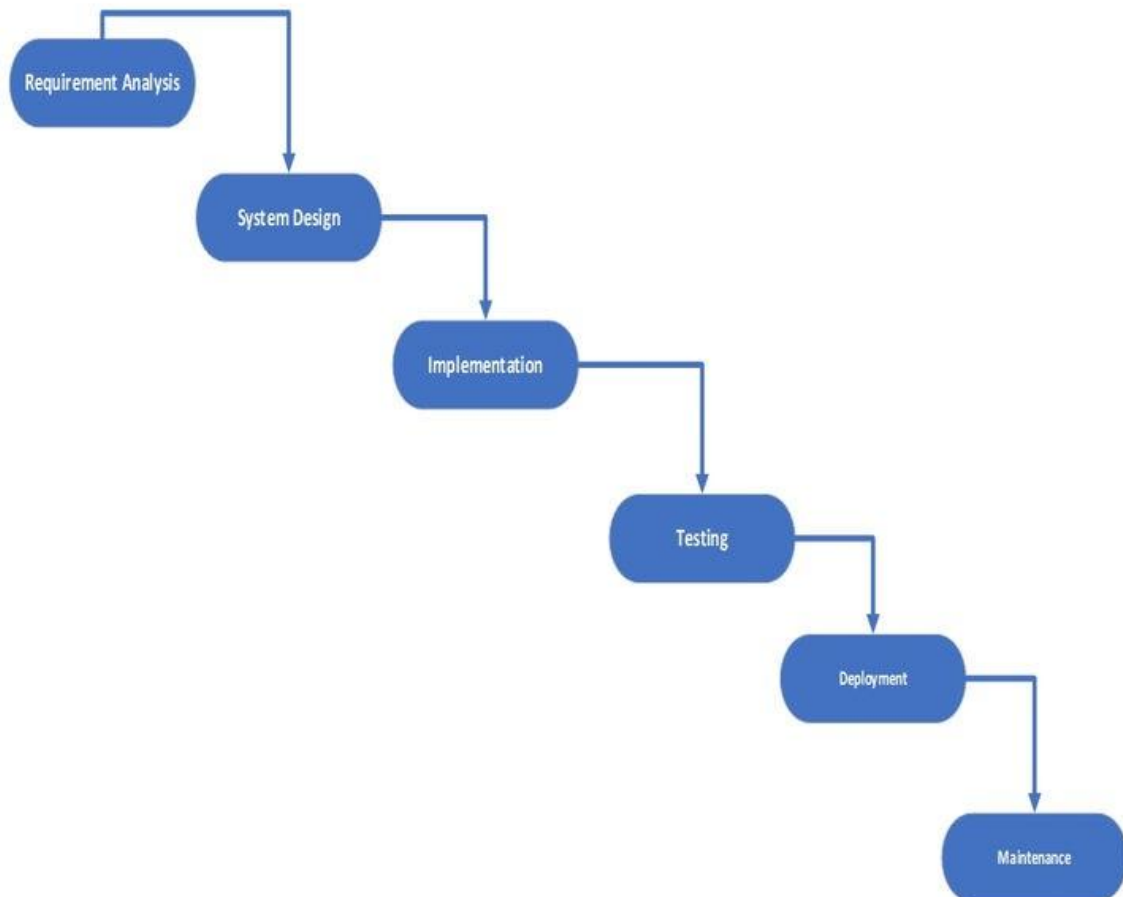
Easy communication

Better data handling

16) Software Development Process

LAB EXERCISE: Create a flowchart representing the Software Development Life Cycle (SDLC).

Ans :



17) Software Requirement

LAB EXERCISE: Write a requirement specification for a simple library management system.

Ans :

1. Functional Requirements:

Add, update, delete books

Issue and return books

Search books by title or author

Manage student/member records

Generate issue/return reports

2. Non-Functional Requirements:

Easy-to-use interface

Fast search performance

Secure login for admin/staff

Data backup support

18) Software Analysis

LAB EXERCISE: Perform a functional analysis for an online shopping system.

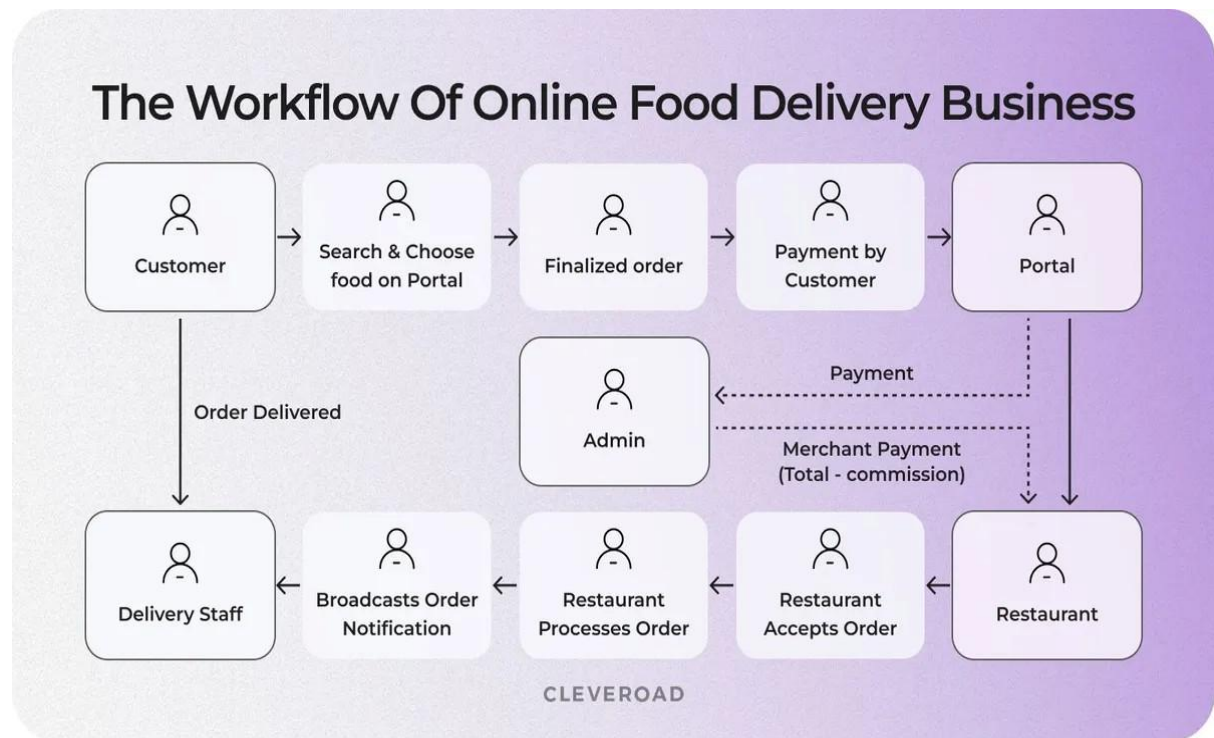
Ans:

1. User Registration/Login
2. Browse Products
3. Add to Cart
4. Place Order
5. Make Payment
6. View Order History
7. Admin Panel to Manage Products

19) System Design

LAB EXERCISE: Design a basic system architecture for a food delivery app.

Ans:



20) Software Testing

LAB EXERCISE: Develop test cases for a simple calculator program.

Ans:

Test Case No.	Input	Operation	Expected Output
TC1	5, 3	Addition (+)	8
TC2	10, 4	Subtraction (-)	6
TC3	6, 7	Multiplication (×)	42
TC4	20, 4	Division (÷)	5
TC5	9, 0	Division (÷)	Error / Cannot divide by zero
TC6	a, 3	Addition (+)	Error / Invalid input
TC7	-5, -2	Addition (+)	-7
TC8	0, 0	Multiplication	0

21) Maintenance

LAB EXERCISE: Document a real-world case where a software application required critical maintenance.

Ans :

Case: WhatsApp Outage (Oct 2022)

Issue: Users couldn't send/receive messages due to server error.

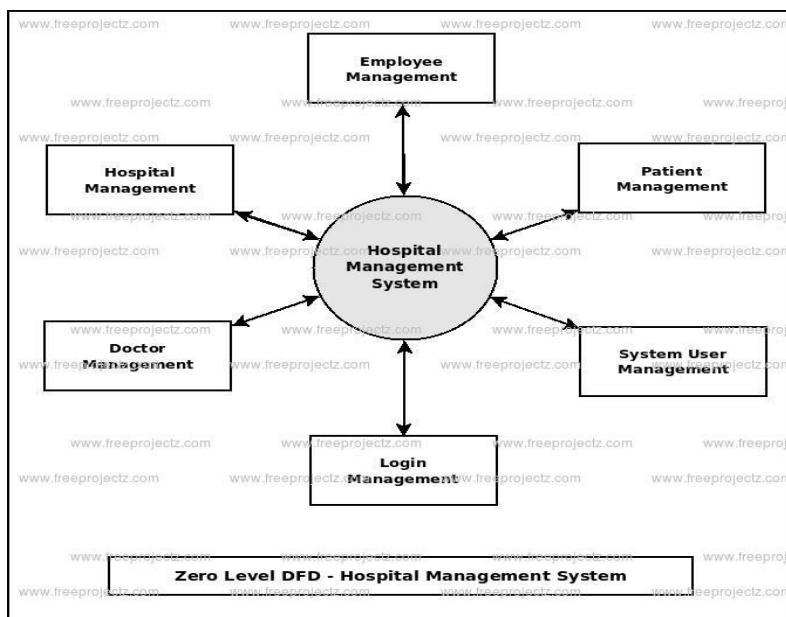
Action: Engineers fixed the backend issue quickly.

Reason: To restore global communication and avoid user disruption.

22) DFD (Data Flow Diagram)

LAB EXERCISE: Create a DFD for a hospital management system.

Ans:



23) Desktop Application

LAB EXERCISE: Build a simple desktop calculator application using a GUI library.

Ans:

```
import tkinter as tk
```

```
def click(event):
```

```
text = event.widget["text"]

if text == "=":

    try:

        result = eval(entry.get())

        entry.delete(0, tk.END)

        entry.insert(0, str(result))

    except:

        entry.delete(0, tk.END)

        entry.insert(0, "Error")

elif text == "C":

    entry.delete(0, tk.END)

else:

    entry.insert(tk.END, text)
```

```
# Main window
```

```
win = tk.Tk()
```

```
win.title("Calculator")
```

```
win.geometry("250x300")
```

```
# Entry box
```

```
entry = tk.Entry(win, font=("Arial", 20))
```

```
entry.pack(pady=10, padx=10, fill="both")
```

```
# Buttons
```

```
buttons = [
```

```
    ["7", "8", "9", "+"],
```

```
    ["4", "5", "6", "-"],
```

```
    ["1", "2", "3", "*"],
```

```
["C", "0", "=", "/"]  
]
```

for row in buttons:

```
    frame = tk.Frame(win)
```

```
    frame.pack(expand=True, fill="both")
```

for btn in row:

```
    b = tk.Button(frame, text=btn, font=("Arial", 16))
```

```
    b.pack(side="left", expand=True, fill="both")
```

```
    b.bind("<Button-1>", click)
```

```
win.mainloop()
```

24) Flow Chart

LAB EXERCISE: Draw a flowchart representing the logic of a basic online registration system.

Ans :

