ASSIGNMENT: PRACITCAL

MODULE 1 – OVERVIEW OF IT INDUSTRY

➥ 1, Write a simple "Hello World" program in two different programming languages of your choice. Compare the structure and syntax.

➥ Proram python 1

```
print("hello,word")
```

OUTPUT:-

```
Hello,world
```

➥

Program html 2

```
<!DOCTYPE html>
<html lang="en">
<head>
    <h1>hello world</h1>
</head>
<body>
</body>
</html>
```

OUTPUT:-

2. Data Transmission: Client to Server

➡

Diagram:

css
CopyEdit
[Client Browser] → [Internet via ISP] → [DNS Lookup] → [Server IP Found]
→ [TCP/IP Handshake] → [HTTP Request] → [Web Server] → [Database]
→ [Response] → [Client Browser]

3. HTTP Client-Server Communication (Python Example)

➡

Server (Flask):

```
python
CopyEdit
from flask import Flask
app = Flask(__name__)

@app.route('/')
def home():
   return "Hello from the server!"

app.run(port=5000)
```

Client (requests):

```
python
CopyEdit
import requests
response = requests.get("http://localhost:5000/")
print(response.text)
```

4. Internet Connection Types: Pros and Cons

➡

Type	Pros	Cons
Broadband	Widely available, decent speed	Speed may drop with more users
Fiber	Extremely fast, reliable	Expensive, limited in rural areas
Satellite	Available in remote areas	High latency, weather sensitive

Type	Pros	Cons
DSL	Uses existing phone lines	Slower compared to modern options
Mobile	Wireless, portable	Data caps, speed depends on signal

5. HTTP and FTP Requests (Using curl)

➥

- **HTTP:**

bash
CopyEdit
curl http://example.com

- **FTP:**

bash
CopyEdit
curl -u username:password ftp://ftp.example.com/file.txt

6. Application Security Vulnerabilities

➥

Vulnerability	Explanation	Solution
SQL Injection	Malicious SQL in input	Use parameterized queries
Cross-Site Scripting	Injecting scripts into web pages	Sanitize and encode user input
Insecure Authentication	Weak login logic	Use strong encryption + token auth

7. Classify 5 Applications

Application	Type

Application	Type
Microsoft Word	Application
Google Chrome	Application
Windows 10	System Software
Antivirus Software	Utility Software
VLC Media Player	Application

8. Three-Tier Web Architecture Diagram

➡ **css**

CopyEdit

Presentation Layer → Business Logic Layer → Data Access Layer
[HTML/CSS/JS] → [Python/Java Logic] → [SQL DB]

9. Case Study: Online Shopping System

➡

- **Presentation Layer**: User interface with product listings, cart, login.
- **Business Logic Layer**: Handles orders, payment processing, discount logic.
- **Data Access Layer**: Manages product, order, and user databases via SQL.

10. Software Environments Setup

➡

Type	Purpose
Development	Code writing and unit testing
Testing	Simulate bugs and QA testing
Production	Live, stable environment

VM Setup: Use tools like VirtualBox with Ubuntu + Python + MySQL.

11. Upload Source Code to GitHub

➡ **bash**

CopyEdit
git init
git add .
git commit -m "Initial commit"
git remote add origin <repo-url>
git push -u origin main

12. Create GitHub Repo & Document Push

➥

- Go to GitHub > New Repository
- Copy repo URL
- Use the Git commands shown above

13. Student GitHub Collaboration

➥

- Both users fork or clone a shared repo.
- One creates issues or tasks.
- Others create branches, make changes, and merge via pull requests.

14. Classify Software

➥

Software	Type
Windows	System Software
MS Office	Application
Disk Cleanup Tool	Utility
Chrome Browser	Application
Task Manager	Utility

15. Git Tutorial: Cloning & Branching

➡

bash
CopyEdit
git clone <url>
git checkout -b feature-branch
git merge feature-branch

16. Report: Types of Application Software

➡

Type	Examples	Productivity Impact
Word Processors	MS Word, Google Docs	Speeds up documentation
Spreadsheets	Excel, Google Sheets	Data analysis and planning
Media Players	VLC, Windows Media	Access to visual/audio info
Browsers	Chrome, Firefox	Internet access, research

17. SDLC Flowchart

➡

css
CopyEdit
[Requirement] → [Design] → [Implementation] → [Testing] → [Deployment] → [Maintenance]

18. Requirement Specification: Library System

➡

Features:

- Add/Delete Books
- Register Users
- Issue/Return Books
- Search Catalog

Functional Requirements:

- Login system
- Search interface
- Fine calculation

19. Functional Analysis: Online Shopping

➡

Functions:

- Register/Login
- Browse Products
- Add to Cart
- Payment Gateway
- Order Confirmation

20. System Architecture: Food Delivery App

➡

- **Client App** (Flutter/React Native)
- **Backend** (Node.js/Express)
- **Database** (MongoDB)
- **APIs** (Payment, Maps)

21. Test Cases: Calculator

➡

Test Case	Expected Output
2 + 2	4
9 / 0	Error/Infinity
Negative number square	Positive result
Clear screen	Reset all input

22. Software Maintenance Case

➥

Case: Aadhaar App Update (India)

- Critical bug caused app crash
- Fixed in emergency patch within 48 hours
- Lesson: Real-time monitoring and logs are crucial

23. DFD: Hospital Management System

➥

Level 0 Diagram:

css

CopyEdit

[Patient] → [Reception] → [Doctor Module] → [Database]

↘→ [Billing System]

24. Desktop Calculator Application (Python + Tkinter)

➥ **python**

CopyEdit

```
import tkinter as tk
root = tk.Tk()
# Add buttons and entry fields
root.mainloop()
```

25. Flowchart: Online Registration System

➥

css

CopyEdit

[Start] → [Enter Details] → [Validate] → [Store in DB]

↘ Invalid → [Show Error] → [Retry]

Tirth Limbachiya