Module 3 – Fundamentals of IT

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

- I'll create a "Hello World" program in Python and C.
- **Python Code:**
- print("Hello, World!")
- **C Code:**
- #include <stdio.h>
-
- int main() {
-     printf("Hello, World!\n");
-     return 0;
- }

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

- Diagram description:
  - Client sends an HTTP request to the server.
  - The server processes the request.
  - The server sends back a response (e.g., HTML page).

    - Data travels through the internet via routers and switches.

3. Design a simple HTTP client-server communication in any language.

- Using Python's http.server and requests libraries:
    - **Server Code:**
    - from http.server import SimpleHTTPRequestHandler, HTTPServer
    -
    - def run():
    -     server_address = ('', 8080)
    -     httpd = HTTPServer(server_address, SimpleHTTPRequestHandler)
    -     print("Server running on port 8080")
    -     httpd.serve_forever()
    -
    - if __name__ == "__main__":
    -     run()
    - **Client Code:**
    - import requests
    - response = requests.get('http://localhost:8080')
    - print(response.text)

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

- **Broadband:**
  - **Pros:** Affordable, widely available, decent speed.
  - **Cons:** Slower speeds compared to fiber.
- **Fiber-optic:**
  - **Pros:** Extremely fast speeds, reliable.
  - **Cons:** Expensive, limited availability.
- **Satellite:**
  - **Pros:** Available in remote areas.
  - **Cons:** High latency, weather-dependent.

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

- **HTTP Request using curl:**
- curl http://example.com
- **FTP Request using curl:**
- curl -u username:password ftp://example.com/file.txt

6. Identify and explain three common application security vulnerabilities. Suggest possible solutions.

- **SQL Injection:**
  - **Solution:** Use parameterized queries or ORM.
- **Cross-Site Scripting (XSS):**

    - **Solution:** Sanitize user inputs and use Content Security Policy (CSP).
- **Cross-Site Request Forgery (CSRF):**
    - **Solution:** Use anti-CSRF tokens in forms.

7. Identify and classify 5 applications you use daily as either system software or application software.

- **System Software:**
    - Operating system (e.g., Windows)
- **Application Software:**
    - Web browser (e.g., Chrome)
    - Word processor (e.g., Microsoft Word)
    - Media player (e.g., VLC)
    - Email client (e.g., Outlook)

8. Create a Github repository and document how to commit and push code changes.

- **Steps:**
    1. Create a new repository on GitHub.
    2. Clone the repository to your local machine:
    3. git clone https://github.com/your-username/repository-name.git
    4. Add changes to the files.
    5. Stage the changes:
    6. git add .

7. Commit the changes:
8. git commit -m "Added new feature"
9. Push the changes:
10. git push origin main

9. Create a student account on Github and collaborate on a small project with a classmate.

- **Steps:**
  1. Sign up on GitHub.
  2. Create a repository for the project.
  3. Invite your classmate to collaborate by adding them as a collaborator.
  4. Work on the project by creating branches, making commits, and merging.

10. Follow a GIT tutorial to practice cloning, branching, and merging repositories.

- **Steps:**
  1. Clone a repository:
  2. git clone https://github.com/your-username/repository-name.git
  3. Create a new branch:
  4. git checkout -b new-feature
  5. Make changes, commit them, and push the branch.
  6. Create a pull request on GitHub.

7. Merge the branch into the main branch after review.

11. Create a DFD (Data Flow Diagram) for a hospital management system.

- Diagram description:
  - **External Entities:** Patients, Doctors, Admin.
  - **Processes:** Patient Registration, Appointment Scheduling, Medical Record Management.
  - **Data Stores:** Patient Database, Appointment Database, Medical Records.

12. Build a simple desktop calculator application using a GUI library.

- **Python Tkinter Code:**

```
import tkinter as tk

def click_button(value):
    current = entry.get()
    entry.delete(0, tk.END)
    entry.insert(tk.END, current + value)

def clear():
    entry.delete(0, tk.END)

```

```
•  def calculate():
•      try:
•          result = eval(entry.get())
•          entry.delete(0, tk.END)
•          entry.insert(tk.END, result)
•      except:
•          entry.delete(0, tk.END)
•          entry.insert(tk.END, "Error")
•
•  root = tk.Tk()
•  root.title("Calculator")
•
•  entry = tk.Entry(root, width=20, font=("Arial", 16))
•  entry.grid(row=0, column=0, columnspan=4)
•
•  buttons = [
•      '7', '8', '9', '/',
•      '4', '5', '6', '*',
•      '1', '2', '3', '-',
•      'C', '0', '=', '+'
•  ]
•
```

```
row = 1
col = 0
for button in buttons:
    if button == "=":
        tk.Button(root, text=button, width=5, height=2, command=calculate).grid(row=row, column=col)
    elif button == "C":
        tk.Button(root, text=button, width=5, height=2, command=clear).grid(row=row, column=col)
    else:
        tk.Button(root, text=button, width=5, height=2, command=lambda value=button: click_button(value)).grid(row=row, column=col)
    col += 1
    if col > 3:
        col = 0
        row += 1

root.mainloop()
```

13. Draw a flowchart representing the logic of a basic online registration system.

- **Flowchart Description:**

    - **Start** → Display Registration Form → Validate User Inputs → Save Data to Database → Send Confirmation Email → **End**

14. Build a simple system architecture for a food delivery app.

- **Components:**
    - **Client Side:** User interface for customers, order management.
    - **Server Side:** API for order processing, payment gateway integration.
    - **Database:** Stores user data, orders, and restaurant menus.

This is a comprehensive list of the **LAB EXERCISES** from the provided content. Each of them involves performing specific tasks or researching topics, as per the instructions. Let me know if you need any additional details or assistance with these exercises!