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

-> python:

Print (“Hello World”);

C:

Include<stdio.h>

Void main ()

{

Printf (“Hello World”);

}

-> C:

-requires headers function, return statement.

-semi colon is required at the end of statement.

-needs to be compiled before running.

-> python:

-no boilerplate needed, one line is enough.

-semi colon is not used.

-interpreted directly by the python interpreter.

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

->

A diagram of a server

AI-generated content may be incorrect.

3.) Research different types of internet connections and list their pros and cons.

-> There are 3 types of internet connections:

1. satellite

* Pros
* Available almost anywhere
* Good for rural or remote areas
* Cons
* High latency
* Weather can affect signal

1. Broadband
   * + Pros

* portable and flexible
* fast speeds
  + - Cons
* Limited by coverage area
* Often comes with data caps

1. Fiber

* Pros
* Fast and reliable
* Ideal for heavy internet usage and multiple users
* Cons
* Expensive installation
* Limited availability in rural areas

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

-> i. SQL injection

- An attacker manipulates SQL queries by injecting malicious input through from fields or URL parameterized.

- solution: use prepared statements and parameterized queries to avoid direct execution of user input.

ii. Cross-Site scripting

-malicious scripts are injected into webpages and executed in user’s browsers.

- solution: sanitize and encode all user inputs and outputs; use content security policy (CSP).

iii. Broken authentication

-Weak login mechanisms allow attackers to compromise user accounts.

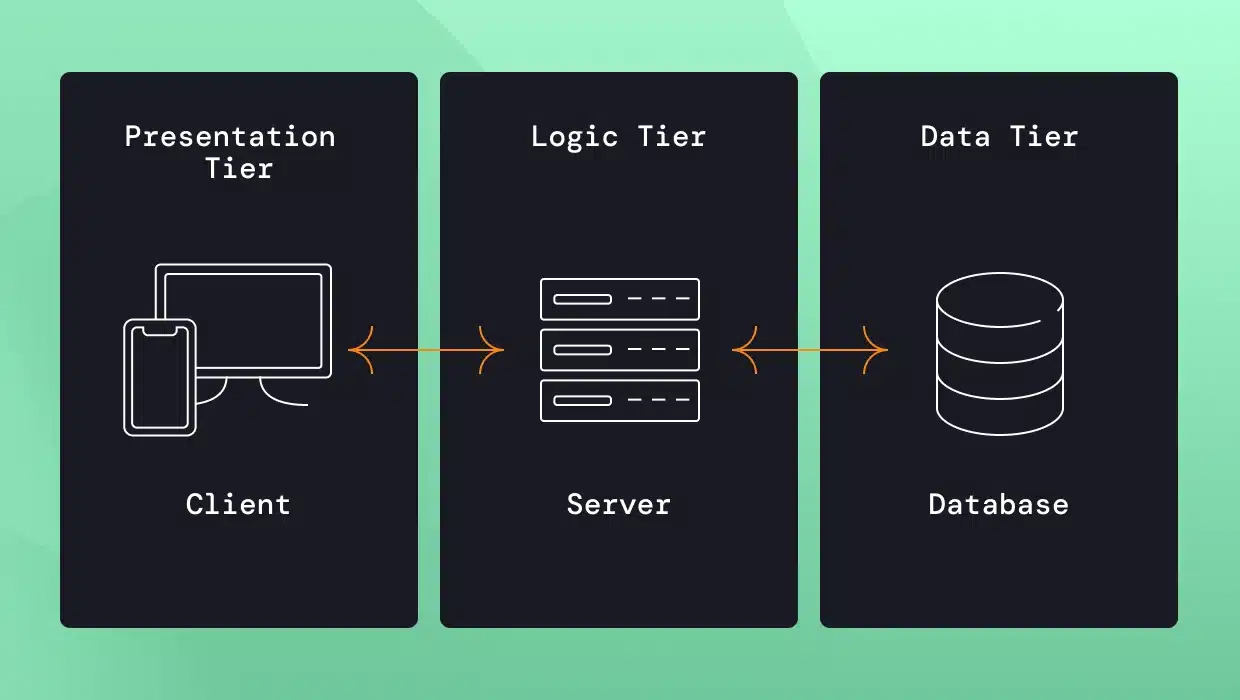
- solution: implement strong password policies, use multi-factor authentication, and securely manage session tokens.

5.) identify and classify 5 application you use daily as either system software or application software.

-> Application software

|  |  |  |
| --- | --- | --- |
| Application name | Type | Classification |
| Google chrome | Web browser | Application software |
| WhatsApp | Messaging App | Application software |
| Instagram | Social media App | Application software |
| Spotify | Music streaming | Application software |
| Calculator | Utility App | Application software |

6.) Design a basic three-tier software architecture diagram for web application.

-> 

7.) Create a case study on the functionality of the presentation, business logic, and dataaccess layers of a given software system.

-> i. Presentation layer

* Function: this is the UI where customers interact with the banking system.
* Technologies: HTML, CSS, JAVASCRIPT mobile app frameworks.
* E.g. login/out screen, error messages and notifications

ii. Business logic layer

* Function: processes user requests, enforces rules, and handles operations.
* Technologies: java, .NET, Python, middleware APIs.
* E.g. validating user credentials, calculating interest or late payments fees

iii. Data access layer

* Function: connects to the database and performs data operations.
* Technologies: SQL, JDBC, ORM tools
* E.g. fetching account details from the database, updating balance after a transfer

9.) explore different types of software environments. Set up a basic environment in virtual machine.

-> 3 types:

1. Development environment

* Purpose: where developers write and test code.
* Characteristics:

- contains code editors, debuggers, compilers.

- fast feedback loop

1. Testing environment

* Purpose: where QA engineers test the application.
* Characteristics:

- Simulates real world scenario using test data.

- used for functional, performance, and regression testing.

1. Production environment

* Purpose: where the application is deployed for end-users.
* Characteristics:

- stable and secure

- uses real user data

* Set up a basic environment in a virtual machine
* Steps:

-> Create the virtual machine

-> Install the OS

-> install development tools

-> Set up web server

-> Simulate a production environment

10.) write and upload your first source code file in GitHub.

-> steps:

1. Write code

* Create ahd.c program in c language using any editor
* ahd.c

#include<stdio.h>

Main ()

{

Printf (“welcome in ahd!!”);

}

1. Create GitHub repository

* Go to github.com -> new repository -> name it -> create

1. Upload file

* In the repository -> click” Add file” -> “upload files”
* Select ahd.c -> click “commit changes”

1. Done

* Share the repository link

11.) Create a GitHub repository and document how to commit and push code changes.

-> steps:

1. Initialize Git

* git init

1. Add Files

* git add.

1. Commit changes

* Git commit – m “your message”

1. Add GitHub repository link

* Git remote add origin

<https://github.com/yourusername/repositoryname.git>

1. Push to GitHub

* Git push – u origin master

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

-> list is below:

* Windows/macOS/Linux - system software
* Google chrome/Safari/Firebox - Application software
* Cloud storage – utility software
* Cleaner apps – utility software
* Mobile banking apps – Application software
* Android/iOS – System software
* WhatsApp – Application software
* YouTube – Application software

13.) follow a GIT tutorial to practice cloning, branching, and merging repositories.

-> steps:

1. Clone a repository

* Git clone <repo\_url>

1. Create a Branch

* Git checkout -b new-branch

1. Switch Branch

* Git checkout main

1. Merge Branch

* Git merge new-branch

1. Push Changes

* Git push origin main

14.) Write a report on the various types of application software and how they improveproductivity.

-> Application software helps users perform specific tasks efficiently. Common types include:

1. Word processors: for creating and editing documents.
2. Spreadsheets: for calculating and data analysis.
3. Presentation: for sharing ideas visually.
4. Databases: for organizing and retrieving data.
5. Graphic software: for designing and editing visuals.

15.) Create a flowchart representing the software Development life cycle (SDLC).

A diagram of a software development process

AI-generated content may be incorrect.

16.) Write a requirement specification for a simple library management System.

-> The library Management system is intended to support the basic operations of a small to medium-sized library.

-> Its primary purpose is to allow librarians to manage book inventories and user accounts, while enabling patrons to search for, borrow, and returns books.

-> The system will consist of two main roles: librarians and patrons.

-> librarians will have administrative privileges, including the ability to add, update, or delete book records, as well as manage user accounts.

17.) Perform a functional analysis for an online shopping system.

-> bellow

* User management
* User registration
* User login/logout
* Profile management
* Password recovery
* Product management
* Product catalog
* Product details page
* Search and filter
* Inventory Management
* Shopping cart
* Add to cart
* View/edit cart
* Price calculation
* Checkout and Payment
* Checkout process
* Payment gateway integration
* Order confirmation
* Order Management
* Order History
* Order tracking
* Admin Order Management
* Reviews and Rating
* Product Reviews
* Moderation
* Promotions and Discounts
* Coupon codes
* Sales and deals
* Customer Support
* Contact form
* Live chat/Help centre
* Security and Privacy
* Data Protection
* Role-based Access
* Secure transactions

18.) Design a basic system architecture for a food delivery app.

-> bellow

1. Overview

* Customers
* Restaurants
* Delivery personnel

1. System Architecture components

* Frontend (client layer)
* Customer App
* Browse restaurants and menus
* Place orders
* Track deliveries
* Manage profile and payments
* Restaurant App
* Manage menu
* Accept/decline orders
* View order history
* Delivery App
* View assigned deliveries
* Navigate to pick-up and drop-off locations
* Update delivery status
* Backend (Application layer)
* User Management Service
* Restaurant Management Service
* Order Management Service
* Delivery Management Service
* Payment Service
* Notification Service
* Review & Rating Service
* Data layer (Database & Storage)
* Relational Database
* NoSQL Database
* Cloud Storage
* External Services
* Payment Gateway integration
* Map/Geolocation API
* Push Notification Services

1. Architectural Diagram (conceptual)

19.) Document a real-world case where a software application required criticalmaintenance.

-> Real -World case:

What was it?

- A major security flaw in OpenSSL, a library used for encrypting internet traffic, due to a missing bound check in the TLS heartbeat extension.

* Impact
* Exposed sensitive data
* Affected millions of websites, including yahoo. Tumblr, and government portals.
* No trace left by attackers, making it hard to detect breaches.
* Critical Maintenance Required
* Immediate patch
* Emergency updates and reboots of servers worldwide.
* SSL certificates revoked and reissued.
* User passwords reset across services.
* Key Takeaways
* Small code bugs can have global consequences.
* Open-source tools need sustained funding and oversight.

20.) create a DFD for a hospital management system.

-> A diagram of a hospital management system

AI-generated content may be incorrect.

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

-> A diagram of a computer network

AI-generated content may be incorrect.