**Frontend Assignment Set**

**Module 1 – Foundation**

**THEORY EXERCISE:**

• What is a HTTP?

**HTTP** stands for **H**yper **T**ext **T**ransfer **P**rotocol

**WWW** is about communication between web **clients** and **servers**

Communication between client computers and web servers is done by sending **HTTP Requests** and receiving **HTTP Responses**

• What is a Browser? How they works?

The web browser is an application software used to explore the World Wide Web [(WWW](https://www.geeksforgeeks.org/world-wide-web-www/)).

It acts as a platform that allows users to access information from the Internet by serving as an interface between the client (user) and the server.

Popular web browsers include [Google Chrome](https://www.geeksforgeeks.org/google-chrome-browser-training-and-support/), [Microsoft Edge](https://www.geeksforgeeks.org/microsoft-edge-browser/), [Mozilla Firefox,](https://www.geeksforgeeks.org/mozilla-firefox-browser/) and [Safari](https://www.geeksforgeeks.org/apple-safari-browser/).

• What is Domain Name?

The **domain Property** is used to return the domain name of the website server that is loaded or running in the current document.

The domain property returns null if the document was created in memory.

• What is hosting?

**Web hosting** is a process and service through which web application or website files are stored in a **Web server** to publish to the Internet via **World Wide Web**. These files are primarily hosted to be publicly available around the world at any time.

**Module 2 – Fundamentals of World Wide Web**

**THEORY EXERCISE:**

• Difference between Web Designer and Web Developer

| **S. No.** | **Web Designer** | **Web Developer** |
| --- | --- | --- |
| 1. | Web Designers are very creative. | Web Developers are more technical. |
| 2. | Web designers refer to how the website looks and feels on the outside. | Web development refers to the functionality of the website on how it works. |
| 3. | Types of Web Designers-   * User Experience (UX) * User Interface (UI) * Visual/Graphic Designer | Types of Web Developers-   * Front-end * Back-end * Full-stack |
| 4. | Web designing tools include Adobe Photoshop, Illustrator, Dreamweaver, Sketch etc. | Languages used are HTML, PHP, JavaScript, CSS, Python, Ruby, jQuery etc. |
| 5. | Most web designers learn about typography and the visual features of a website. | They use web tools and programming languages to create the design and functionality of a website. |

• What is a W3C?

The W3C (World Wide Web Consortium) is an international organization that creates standards for the [World Wide Web](https://www.techtarget.com/whatis/definition/World-Wide-Web). The WC3 is committed to improving the web by setting and promoting web-based standards.

• What is Domain?

A domain is a [network](https://www.webopedia.com/definitions/network/) of [computers](https://www.webopedia.com/definitions/computer/) and [devices](https://www.webopedia.com/definitions/device/) that are controlled by one set authority and have specific guidelines.

• What SEO?

SEO stands for Search Engine Optimization and helps search engines understand your website’s content and connect it with users by delivering relevant, valuable results based on their search queries.

• What is SDLC life cycle?

The software development lifecycle (SDLC) is the cost-effective and time-efficient process that development teams use to design and build high-quality software.

**Module 3 – Fundamentals of IT**

What is a Program?

**THEORY EXERCISE:**

🡪Explain in your own words what a program is and how it functions.

A program is a piece of code or set of [instructions](https://www.webopedia.com/definitions/instruction/) that tells a computer how to perform a task. To use an analogy, a program is like a computer’s recipe. It contains a list of ingredients (called [variables](https://www.webopedia.com/definitions/variable/), which can represent numeric [data](https://www.webopedia.com/definitions/data/), [text](https://www.webopedia.com/definitions/text/), or images) and a list of directions (called [statements](https://www.webopedia.com/definitions/statement/)) that tell the computer how to [execute](https://www.webopedia.com/definitions/execute/) a specific task.

What is Programming?

**THEORY EXERCISE:**

🡪What are the key steps involved in the programming process?

🡪The Key steps in the programming process are:

1) Understanding the problem and defining requirements,

2) Designing a solution,

3) Coding the solution,

4) Testing and debugging the code,

5) Documenting the program and

6) Maintaining the program.

Types of Programming Languages

**THEORY EXERCISE:**

🡪What are the main differences between high-level and low-level programming languages?

| **Parameters** | **High-Level Language** | **Low-Level Language** |
| --- | --- | --- |
| **Ease of Use** | Easier to learn and use | More complex and harder to learn |
| **Portability** | Highly portable across different systems | Less portable, often system-specific |
| **Development Speed** | Faster development time | Slower development time |
| **Examples** | Python, Java, C++, JavaScript | Assembly language, Machine code |
| **Memory Management** | Automatic memory management | Manual memory management |
| **Performance** | Generally slower execution | Generally faster execution |

World Wide Web & How Internet Works

**THEORY EXERCISE:**

🡪Describe the roles of the client and server in web communication.

The client (browser) sends requests to the server to access web pages,

Images, videos, or other resources.

The server sends back the requested data or a response message to the client.

Network Layers on Client and Server

THEORY EXERCISE:

🡪Explain the function of the TCP/IP model and its layers.

The TCP/IP model is a framework that is used to model the communication in a network. It is mainly a collection of network protocols and organization of these protocols in different layers for modeling the network.

### **1. Application Layer**

### **2. Transport Layer**

### **3. Internet Layer**

### **4. Network Access Layer**

Client and Servers

**THEORY EXERCISE:**

🡪Explain Client Server Communication

Client-server communication is a model where a client (e.g., browser, mobile app) requests services from a server (e.g., web server, database), which processes the request and responds over a network using protocols like HTTP, WebSockets. It enables distributed computing, ensuring efficient resource sharing and data exchange.

Types of Internet Connections

**THEORY EXERCISE:**

🡪 How does broadband differ from fiber-optic internet?

Broadband is a type of internet service that provides its customers with high-speed internet access. Broadband is capable of sending multiple signals over a wide bandwidth. Broadband uses optical fibre, coaxial cable, twisted pair, and radio as its medium. It can be used by organisations as well as by individuals. The fibre cables used in fibre optic are used to transmit data through them. Superfast broadband is quicker than standard broadband, and the connection is also more reliable.

Protocols

**THEORY EXERCISE:**

**🡪**What are the differences between HTTP and HTTPS protocols?

| **HTTP** | **HTTPS** |
| --- | --- |
| HTTP stands for Hyper Text Transfer Protocol. | HTTPS for Hyper Text Transfer Protocol Secure. |
| In HTTP, URL begins with “http://”. | In HTTPs, URL starts with “https://”. |
| HTTP uses port number 80 for communication. | HTTPs uses 443 port number for communication. |
| HTTP is considered to be unsecure. | HTTPs is considered as secure. |
| HTTP works at Application Layer. | HTTPS works at Transport Layer. |
| HTTP faster than HTTPS | HTTPS slower than HTTP |

Application Security

**THEORY EXERCISE:**

🡪What is the role of encryption in securing application, Software Applications and Its Types

Encryption is a fundamental component of application security, offering robust protection for sensitive data and ensuring secure communication channels. Encryption transforms plaintext into cipher text using algorithms.

### **1) Symmetric Encryption**

### **2) Asymmetric Encryption**

**THEORY EXERCISE:**

🡪 What is the difference between system software and application software?

| **System Software** | **Application Software** |
| --- | --- |
| System Software maintains the system resources and gives the path for application software to run. | Application software is built for specific tasks. |
| Low-level languages are used to write the system software. | While high-level languages are used to write the application software. |
| It is general-purpose software. | While it's a specific purpose software. |
| Without system software, the system stops and can't run. | While Without application software system always runs. |
| Example: System software is an operating system, etc. | Example: Application software is Photoshop, VLC player, etc. |

Software Architecture

**THEORY EXERCISE:**

🡪What is the significance of modularity in software architecture?

The module simply means the software components that are been created by dividing the software. The software is divided into various components that work together to form a single functioning item but sometimes they can perform as a complete function if not connected with each other. This process of creating software modules is known as **Modularity** in software engineering.

Layers in Software Architecture

**THEORY EXERCISE:**

🡪Why are layers important in software architecture?

Layered architecture structures an application into several layers, each serving a different purpose. These layers interact in a top-down manner, where higher layers depend on lower layers but not vice-versa.

Software Environments

**THEORY EXERCISE:**

🡪Explain the importance of a development environment in software production.

A Software Development Environment (SDE) is a crucial ecosystem for developers, comprising hardware and software tools that facilitate the creation, testing, and deployment of software systems.

Source Code

**THEORY EXERCISE:**

🡪What is the difference between source code and machine code?

1) machine code:-

**Machine code** is a set of binary instructions that are executed directly by a computer’s CPU. It is the lowest level of programming language and is specific to the architecture of the processor.

2) source code:-

**Source code** is the human-readable form of a program written in a high-level programming language such as C++, Java, or Python. Source code must be compiled or interpreted into machine code before it can be executed by a computer.

Github and Introductions

**THEORY EXERCISE:**

🡪 Why is version control important in software development?

Version Control Systems (VCS) are essential tools used in software development and collaborative projects to track and manage changes to code, documents and other files.

Student Account in Github

**THEORY EXERCISE:**

🡪What are the benefits of using Github for students?

GitHub is a popular online platform where developers can host and share their code. GitHub is an essential tool for developers, and and it’s used by some of the largest technology companies in the world use it.

Types of Software

**THEORY EXERCISE:**

🡪What are the differences between open-source and proprietary software?

| **S.No.** | **OPEN-SOURCE SOFTWARE** | **PROPRIETARY SOFTWARE** |
| --- | --- | --- |
| **1.** | In open-source software the source code is public. | In proprietary software, the source code is protected. |
| **2.** | Open-source software is managed by an open-source community of developers. | Proprietary software is managed by a closed team of individuals or groups that developed it. |
| **3.** | Open-source software can be installed on any computer. | Proprietary software can not be installed into any computer without a valid license. |
| **4.** | Examples are Android, Linux, Firefox, Open Office, VLC Media player, etc. | Examples are Windows, mac OS, Internet Explorer, Google Earth, Microsoft Office, Adobe Flash Player, etc. |

GIT and GITHUB Training

**THEORY EXERCISE:**

🡪How does GIT improve collaboration in a software development team?

When multiple developers are working on the same project, Git collaboration plays an important role in providing easy coordination and efficient code management.

Application Software

**THEORY EXERCISE:**

🡪What is the role of application software in businesses?

Business application software refers to specialized computer programs that are built to optimize and support various operational processes within a business.

Software Development Process

**THEORY EXERCISE:**

🡪What are the main stages of the software development process?

The software development process is the approach to developing, and delivering software applications.

Software Requirement

**THEORY EXERCISE:**

🡪Why is the requirement analysis phase critical in software development?

Requirement analysis is a crucial stage in software development where the needs and expectations of stakeholders are identified and documented.

Software Analysis

**THEORY EXERCISE:**

🡪What is the role of software analysis in the development process?

Software analysis and design tools are essential for developing high-quality software systems.

System Design

**THEORY EXERCISE:**

🡪What are the key elements of system design?

The process of specifying a computer system's architecture, components, modules, interfaces, and data is known as system design.

Software Testing

**THEORY EXERCISE:**

🡪Why is software testing important?

Software testing is essential to ensure that applications work correctly, perform well, and are free from critical bugs before reaching users.

Maintenance

**THEORY EXERCISE:**

🡪What types of software maintenance are there?

🡪The four types are:  
Corrective Software Maintenance  
Preventative Software Maintenance   
Perfective Software Maintenance  
Adaptive Software Maintenance

Development

**THEORY EXERCISE:**

🡪What are the key differences between web and desktop applications?

| **Point** | **Web-based Application** | **Desktop Application** |
| --- | --- | --- |
| **Installation and Setup** | No installation needed; accessed through a browser | Requires installation on a computer or device |
| **Internet Requirement** | Needs a stable internet connection | Can work offline after installation |
| **Performance and Speed** | Depends on server response and browser speed | Uses local device resources for faster performance |
| **Updates and Maintenance** | Centralized updates handled by the server | Manual or automatic updates on each device |

Web Application

**THEORY EXERCISE:**

🡪What are the advantages of using web applications over desktop applications?

## ****Web App Advantages****

### No Downloading and Installation

### Cross-Platform Compatibility

## Desktop App Advantages

### Offline Access

### More Privacy

Designing

**THEORY EXERCISE:**

🡪What role does UI/UX design play in application development?

UI/UX Design, referred to as “User Interface/User Experience design,” has become very important in the field of application design focusing on creating a visual interface and improving the experience of the user.

Mobile Application

**THEORY EXERCISE:**

🡪What are the differences between native and hybrid mobile apps?

|  |  | **s.no** | **Native Apps** | **Hybrid Apps** |
| --- | --- | --- | --- | --- |
|  |  | **Development speed** | Slow | Fast |
|  |  | **Maintenance cost** | High | Low |
|  |  | **Graphical Performance** | Very High | Moderate |
|  |  | **Language Used** | Kotlin, java, Swift | HTML, CSS, JavaScript |
|  |  | **Code Portability** | Tough | Easy |

DFD (Data Flow Diagram)

**THEORY EXERCISE:**

🡪What is the significance of DFDs in system analysis?

**Data Flow Diagram** is a **visual representation** of the flow of data within the system.

Desktop Application

**THEORY EXERCISE:**

🡪What are the pros and cons of desktop applications compared to web applications?

## ****Web App Advantages****

### No Downloading and Installation

### Cross-Platform Compatibility

## Web Application Disadvantages

### Requirement of Internet

### Slow Performance

## Desktop App Advantages

### Offline Access

### More Privacy

## Desktop App Disadvantages

### More Space Utilization

### Upgraded Manually

Flow Chart

**THEORY EXERCISE:**

🡪How do flowcharts help in programming and system design?

The flowcharts are simple visual tools that help us understand and represent processes very easily.

**Module 2 – Frontend – HTML**

**HTML Basics**

**Theory Assignment**

• Question 1:

🡪 Define HTML. What is the purpose of HTML in web development?

**HTML** is the standard mark up language used to structure web pages.

One of the primary purposes of HTML is to ensure that web content is well-structured, making it easier for browsers to interpret and display the information accurately across different devices and platforms.

• Question 2:

🡪 Explain the basic structure of an HTML document. Identify the mandatory tag sand their purposes.

*<!DOCTYPE html>*

<html>

<head>

*<!-- The head section goes here -->*

</head>

<body>

*<!-- The body section goes here -->*

</body>

</html>

### 1**. <!DOCTYPE html>: Declaring the Document Type**

### 2. **<html>: The Root Element**

### 3. **<head>: Metadata of the Document**

### 4. **<body>: Visible Content of the Webpage**

• Question 3:

🡪 What is the difference between block-level elements and inline elements in HTML? Provide examples of each.

A block-level element always starts on a new line, and the browsers automatically add some space (a margin) before and after the element.

Two commonly used block elements are: <p> and <div>.

An inline element does not start on a new line.

This is a <span> element inside a paragraph.

• Question 4:

🡪Discuss the role of semantic HTML. Why is it important for accessibility and SEO? Provide examples of semantic elements.

HTML5 introduced a range of **semantic elements**that clearly describe their purpose in human and machine-readable language.

* **SEO:**Better structured data leads to better SEO. Search engines prioritize well-structured content that uses semantic elements correctly, as it’s easier to index.

Example of semantic element

<header>, <footer>, <aside>, <nav>, <section>, <main>

**HTML Forms**

**Theory Assignment**

• Question 1:

🡪 What are HTML forms used for? Describe the purpose of the input, text area, select, and button elements.

HTML forms, defined using the **<form> Tags** are essential for collecting user input on web pages.

| **Elements** | **Descriptions** |
| --- | --- |
| [**<input>**](https://www.geeksforgeeks.org/html/html-input-tag/) | It is used to get input data from various types such as text, password, email, etc by changing its type. |
| [**<text area>**](https://www.geeksforgeeks.org/html/html-textarea-name-attribute/) | It is used to get input long text content. |
| [**<select>**](https://www.geeksforgeeks.org/html/html-select-tag/) | It is used to create a drop-down list. |
| [**<button>**](https://www.geeksforgeeks.org/html/button-tag-vs-input-typebutton-attribute/) | It defines a clickable button to control other elements or execute a functionality |

• Question 2:

🡪Explain the difference between the GET and POST methods in form submission. When should each be used?

|  |  |
| --- | --- |
| **HTTP GET** | **HTTP POST** |
| GET request is comparatively better than Post so it is used more than the Post request. | POST request is comparatively less better than Get method, so it is used less than the Get request. |
| GET requests are only used to request data (not modify) | POST requests can be used to create and modify data. |
| GET method request can be saved as bookmark in browser. | POST method request can not be saved as bookmark in browser. |
| Request made through GET method are stored in Browser history. | Request made through POST method is not stored in Browser history. |

Question 3:

🡪What is the purpose of the label element in a form, and how does it improve accessibility?

The <label> tag defines a label for many form elements.

The <label> element is useful for screen-reader users, because the screen-reader will read out loud the label when the user focuses on the input element.

**HTML Tables**

**Theory Assignment**

• Question 1:

🡪Explain the structure of an HTML table and the purpose of each of the following elements:

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | |  |  | |  |

<table>, <tr>, <tr>, <td>, and<thead>

|  |  |
| --- | --- |
| **Tag** | **Description** |
| [<table>](https://www.w3schools.com/tags/tag_table.asp) | Defines a table |
| [<th>](https://www.w3schools.com/tags/tag_th.asp) | Defines a header cell in a table |
| [<tr>](https://www.w3schools.com/tags/tag_tr.asp) | Defines a row in a table |
| [<td>](https://www.w3schools.com/tags/tag_td.asp) | Defines a cell in a table |

|  |  |
| --- | --- |
| [<thead>](https://www.w3schools.com/tags/tag_thead.asp) | Groups the header content in a table |

<table>  
  <tr>  
    <th>Company</th>  
    <th>Contact</th>  
    <th>Country</th>  
  </tr>  
  <tr>  
    <td>Alfreds Futterkiste</td>  
    <td>Maria Anders</td>  
    <td>Germany</td>  
  </tr>  
  <tr>  
    <td>Centro comercial Moctezuma</td>  
    <td>Francisco Chang</td>  
    <td>Mexico</td>  
  </tr>  
</table>

• Question 2:

🡪What is the difference between colspan and rowspan in tables? Provide examples.

| **S. No.** | **Row span** | **Col span** |
| --- | --- | --- |
| 1. | ROWSPAN is used to combine the cells vertically. It merges the number of cells vertically and displays them as a single cell. | COLSPAN is used to combine the cells horizontally. It specifies the number of columns that the cells span across and shows them as a single cell. |
| 2. | E.g. <TD Row span= "3"> where 3 is the number of rows that the cells span across. | E.g. <TD Col span= "4"> where 4 is the number of columns that the cells span across. |

• Question 3:

🡪Why should tables be used sparingly for layout purposes? What is a better alternative?

A better alternative for web page layout is to use CSS (Cascading Style Sheets) in conjunction with semantic HTML elements.

* **Semantic HTML:**

Utilize elements like div, header, nav, main, article, section, aside, and footer to structure the content logically.

* **CSS for Styling and Layout:**

Employ CSS properties and layout models like:

* + **Flexbox:** For one-dimensional layouts (rows or columns) and distributing space among items.
  + **CSS Grid:** For two-dimensional layouts, creating complex grid structures with precise control over rows and columns.
  + **Other CSS properties:** float, position, display, margin, padding, etc., for fine-tuning element placement and appearance.

This approach offers superior flexibility, accessibility, maintainability, and responsiveness compared to table-based layouts.