# Frontend Assignment Set

Module 1 – Foundation

**THEORY EXERCISE**:

1. What is a HTTP?

* **HTTP** stands for **Hyper Text Transfer Protocol.**
* HTTP is a **language** that web browsers and server used to talk with each other.
  + When you open a website, your browser **sends an HTTP request to server**.
  + The server **sends back an HTTP response**.

2 What is a Browsers? How they work?

A **Web Browser** is a **software application** that allows you to access and view websites on the internet.

Some common Browsers are:

* Google chrome
* Mozilla Firefox
* Microsoft Edge
* Safari
* Opera

3 What is Domain Name?

* A **Domain name** is the **address** of a website that you type into the browser to visit it.
* Every domain name is unique no two websites can have the same domain name.

4 What is hosting?

* Hosting or Web hosting is a **Service** that stores your website’s file (HTML, CSS, Images, etc.) on a **server**, and make them available on the internet.
* Or we can say, if your **Domain name** is address, then **Hosting** is the house where your website lives.
* We can buy a hosting plan from a provider like;

**Hostinger**

**GoDaddy**

**Bluehost**

**AWS, Google Cloud,** etc.

Module 2 – Fundamentals of World Wide Web

**THEORY EXERCISE:**

1. Difference between Web Designer and Web Developer?

|  |  |
| --- | --- |
| **Web Designer** | **Web Developer** |
| * **Focus:**   Appearance and user experience (UX/UI). | * **Focus:**   Functionality and technical implementation. |
| * **Key Skills:**   Graphic design, layout design, color theory, typography, responsive design. | * **Key Skills:**   Coding in HTML, CSS, JavaScript, and backend languages (like PHP, Python, Node.js). |
| * **Tools Used:**   Adobe XD, Figma, Sketch, Photoshop, Illustrator. | * **Tools Used:**   Code editors (VS Code), frameworks (React, Angular, Laravel), version control (Git). |
| * **Main Responsibilities:**   + Designing the look and feel of the website.   + Creating wireframes and mockups.   + Ensuring visual consistency and user-friendliness.   + Sometimes using HTML/CSS for front-end styling, but not always required. | * **Main Responsibilities:** * Turning design mockups into functional websites. * Writing the code for website behavior and functionality. * Handling server-side logic and databases (if full-stack or backend developer). * Ensuring website performance, security, and responsiveness. |

1. What is a W3C?

**W3C (World Wide Web Consortium)** is a group that makes rules for how websites should work.

**In simple terms:**

* It was started by **Tim Berners-Lee**, the creator of the web.
* It creates **standards** (rules) for web things like **HTML, CSS**, and **accessibility**.
* These rules help websites look and work the same in all browsers and devices.
* It makes the web **organized, fair, and easy to use** for everyone.

1. What is Domain?

A **domain** is the **name of a website** that people type into a browser to visit it.

**In simple words:**

It’s like the **address** of your house, but for a website.

**Example:**

* In [**www.google.com**](http://www.google.com), the domain is **google.com**.
* It helps users find your website without needing to know its IP address.

1. What is SEO?

**SEO** stands for **Search Engine Optimization**.

**In simple words:**

It’s the process of making your website **show up higher** on search engines like **Google** when people search for something.

**Why it’s important:**

* Higher ranking = More visitors to your website.
* Good SEO = Better visibility and more clicks.

**Basic SEO includes:**

* Using the right **keywords**.
* Writing **useful content**.
* Making your site **fast** and **mobile-friendly**.
* Getting **links** from other trusted websites.

1. What is SDLC life cycle?

**SDLC** stands for **Software Development Life Cycle**.

**In simple words:**

It's a step-by-step process used to **plan, develop, test, and deliver** software.

**Main Stages of SDLC:**

* **Planning** – Decide what the software should do.
* **Analysis** – Understand the requirements and needs.
* **Design** – Plan how the software will look and work.
* **Development** – Write the actual code.
* **Testing** – Check for bugs and fix them.
* **Deployment** – Release the software to users.
* **Maintenance** – Update and improve it after release.

**Why it's useful:**

It helps teams **build software in an organized way**, reduce errors, and finish projects on time.