

Web Application Workshop

Computer Science and Informatics
School of Engineering



**London
South Bank**
University



Lsbu_csi ; lsbu_csi_community



Workshop Plan

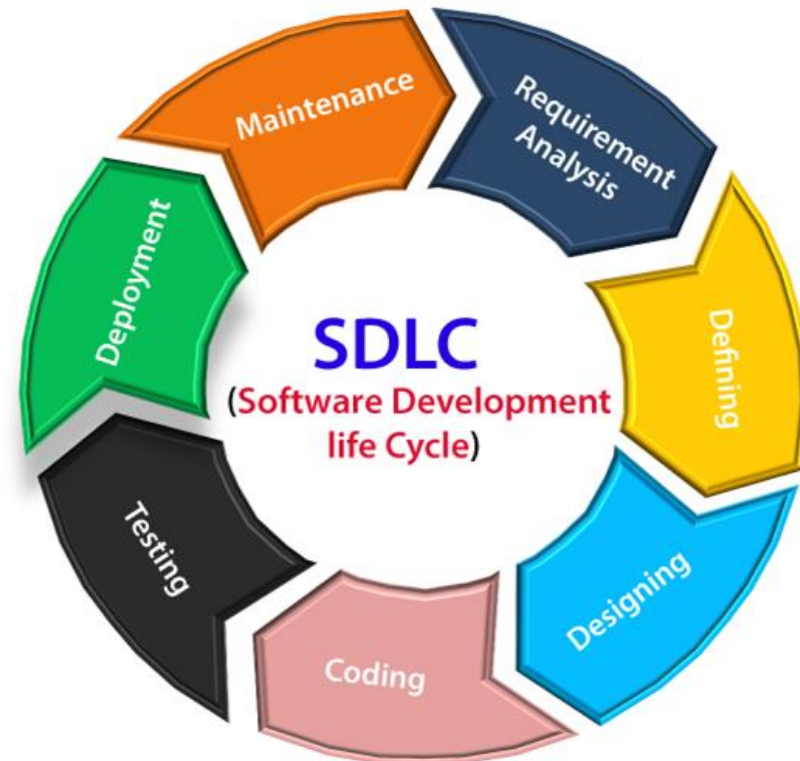
- Session 1:
 - Introduction to Figma
 - Wireframe design of the application.
 - Get Git and GitHub ready
 - Introduction to Web Development
- Session 2- 4
 - Develop the website (HTML CSS)
 - Use Git and GitHub.
 - Database creation (MySQL)
- Session 5
 - Deploy the Website
 - Share on LinkedIn Profile

What is design?

Plan or blueprint of a Software system.

Purpose of Design:

- Clarification of Requirements
- Plan and organise software components
- Feedback and communication
- Areas including, UML design, UI design



Low Fidelity Design

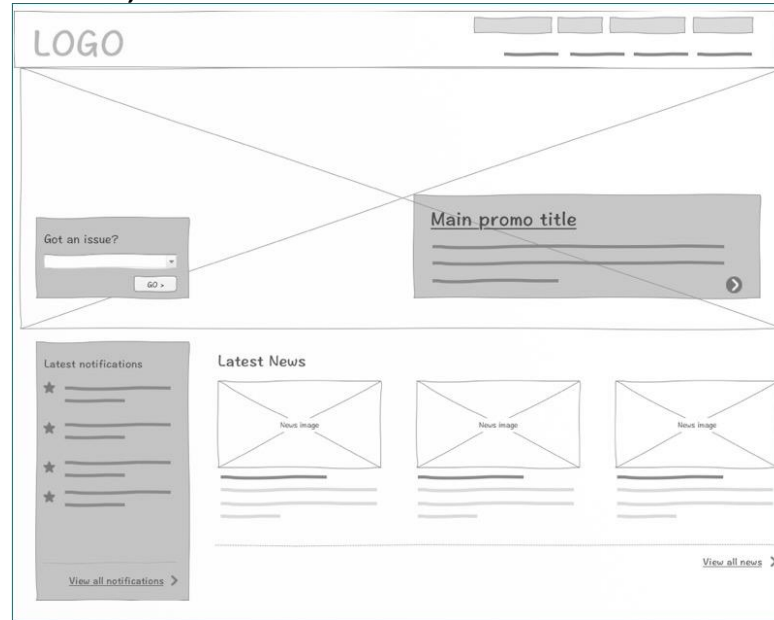
Sketch:



Tools:

- Pen and Paper
- Apps: Procreate, SketchBook, any applications allows drawing

Wireframe: High-level sketch, no detailed visual elements

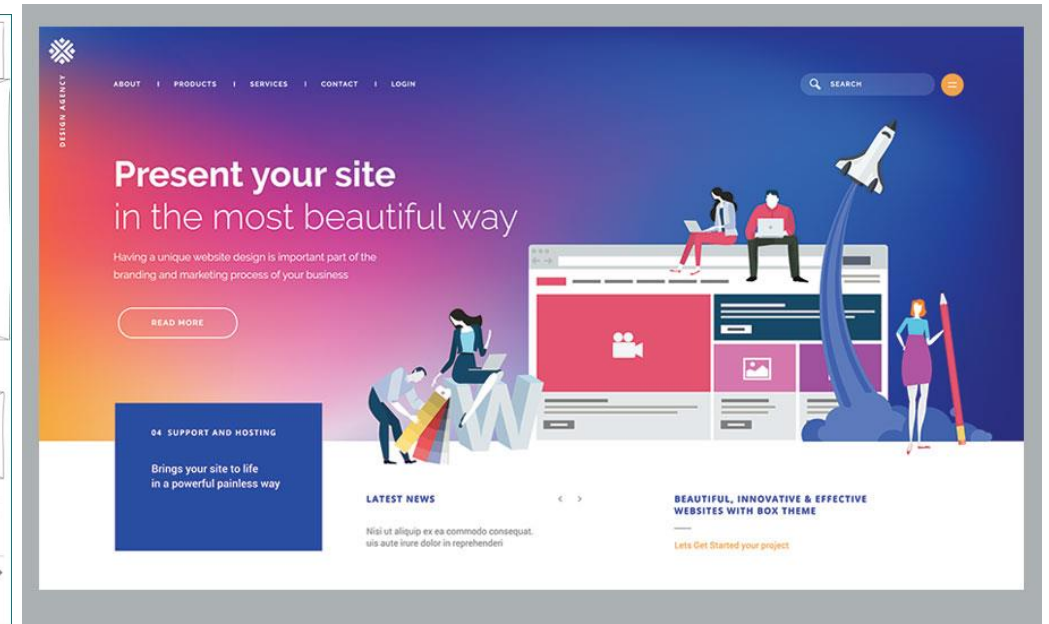


Tools:

- Balsamiq, Figma, Sketch, Mockflow

High Fidelity Design

Mockup: Static visual



Tools:

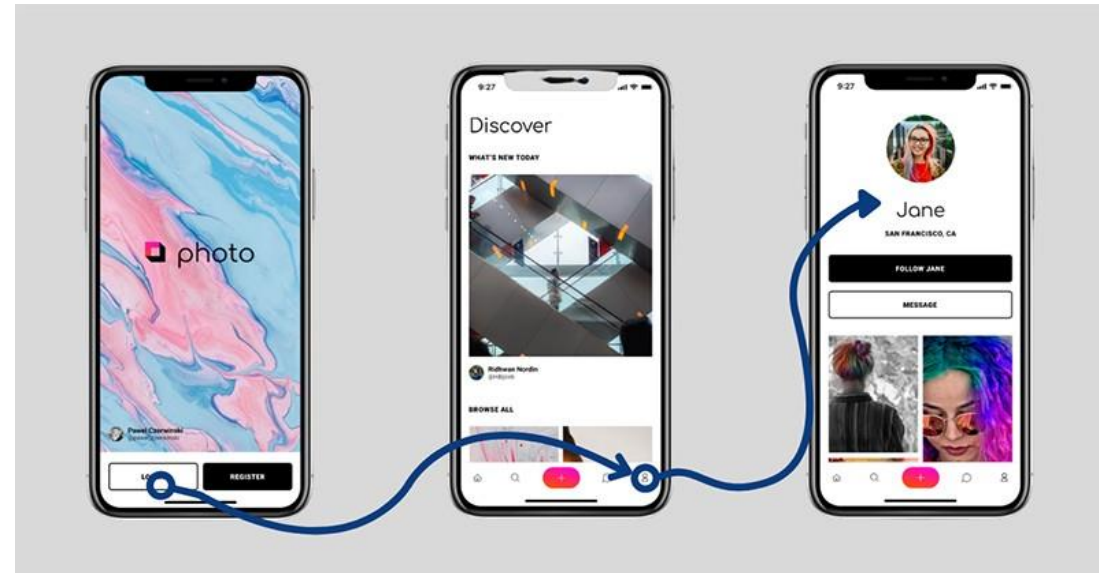
- Balsamiq, Figma, Adobe XD, Sketch, Adobe Photoshop, Zeplin

Prototype

- A dynamic model of the product.
- More interactive with details.

Wireframe/Mockup link of the Quiz App:

<https://www.figma.com/file/tGlAY3WfqdQTTY9TAqfYqC/QuizApp?type=design&node-id=0%3A1&mode=design&t=3H4cGwGWay1mL9I8-1>



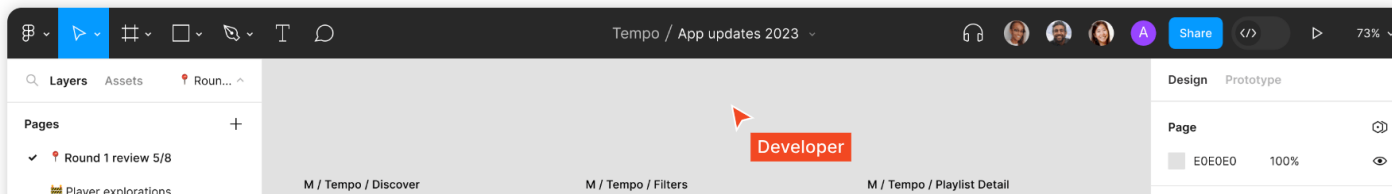
Introduction to Figma

Access from any browser: <https://www.figma.com/>



How you **design** , **align** , and **build**  matters. Do it together with Figma.

Get started



Pricing



Starter

Free

Select plan

- ✓ Figma editor
- ✓ 3 collaborative design files
- ✓ Unlimited personal drafts
- ✓ Basic file inspection

Professional

£11/mo

Dev Mode included

Select plan

Free for students and educators

Everything in Starter and...

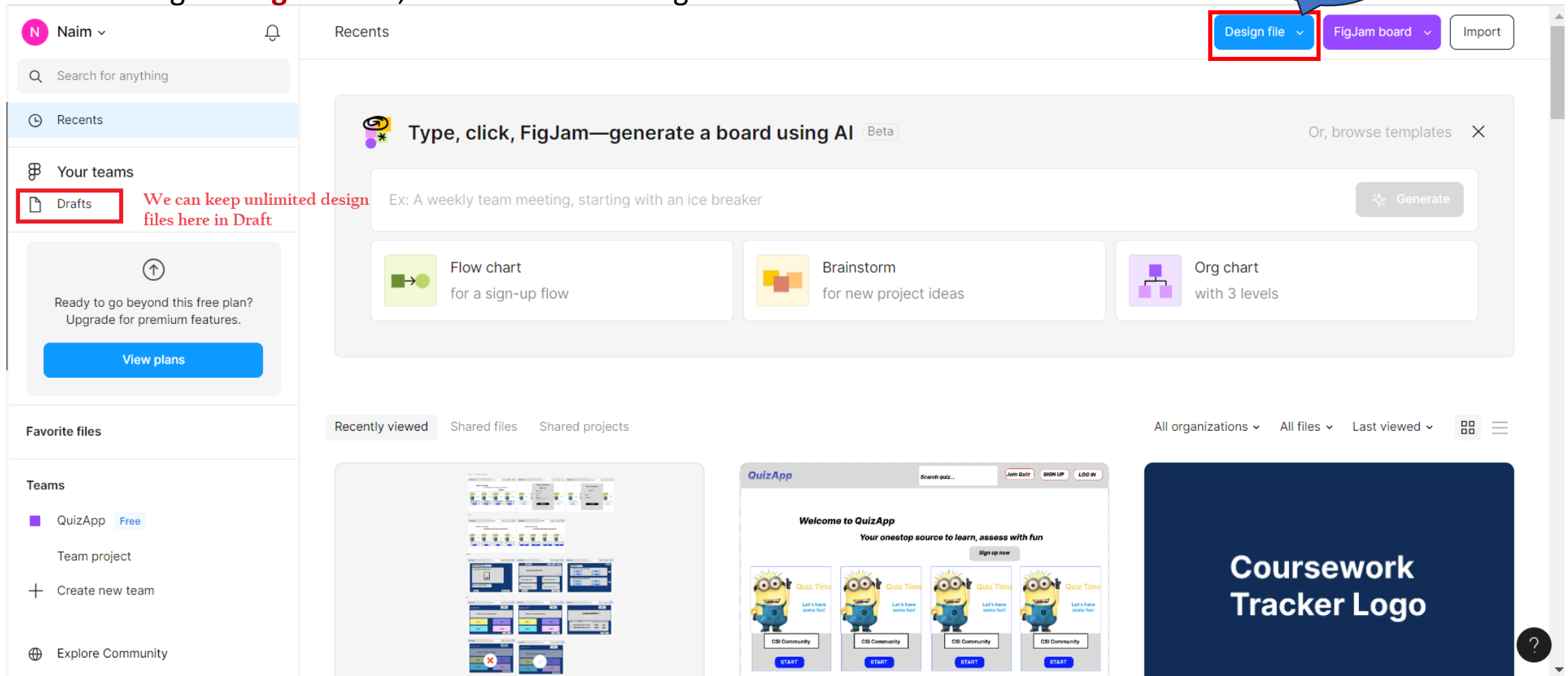
Design

- ✓ Unlimited Figma files
- ✓ Team libraries
- ✓ Advanced prototyping

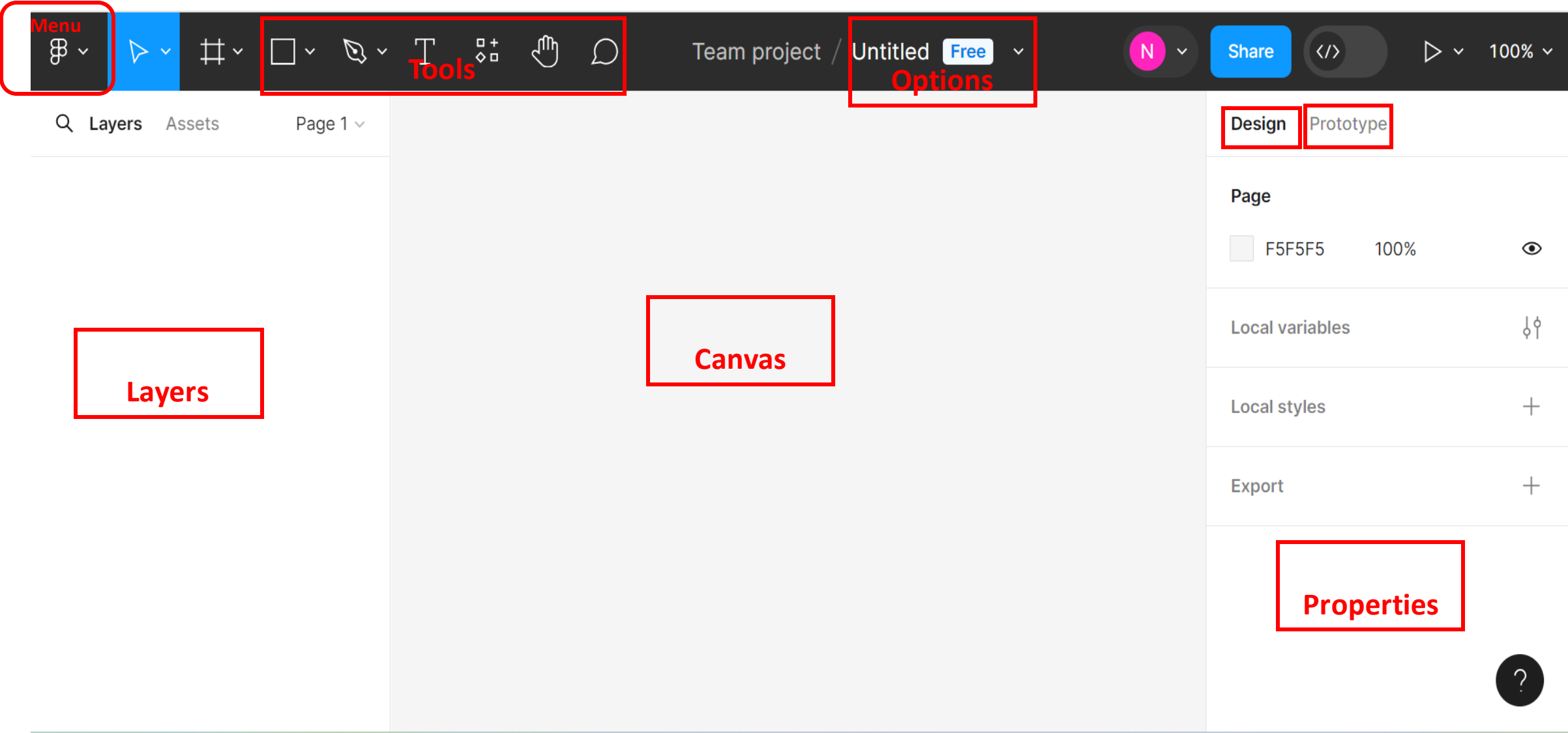
Dev Mode

Introduction to Figma

Once we login to **figma.com**, we see the following screen.

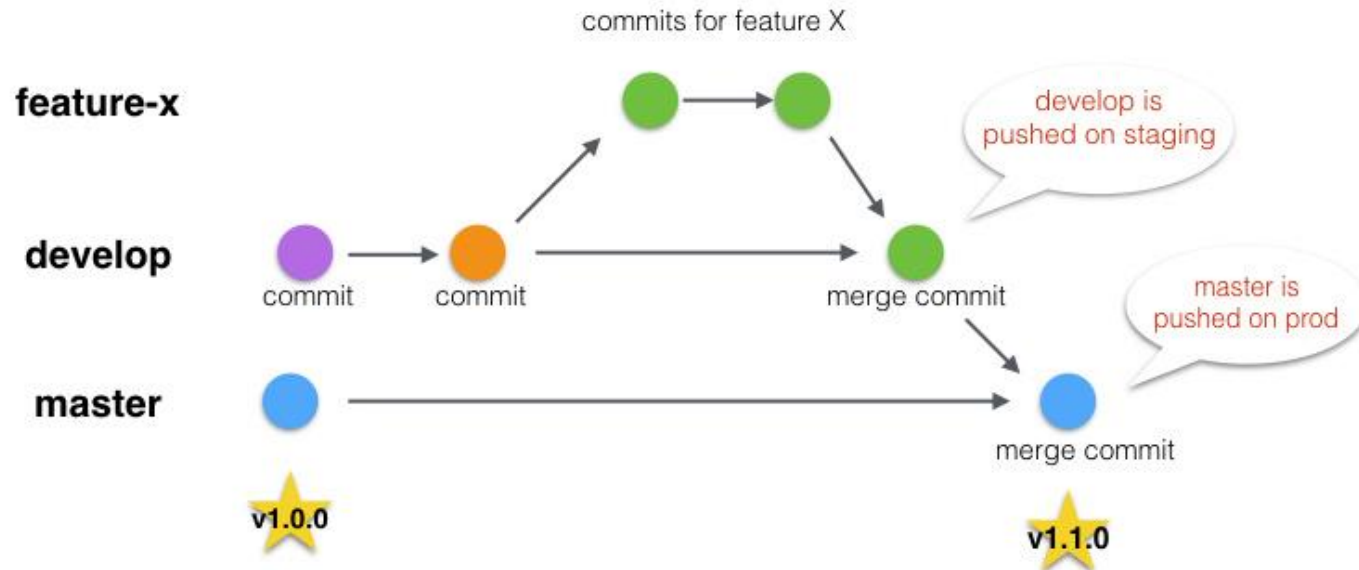


Figma Design Interface

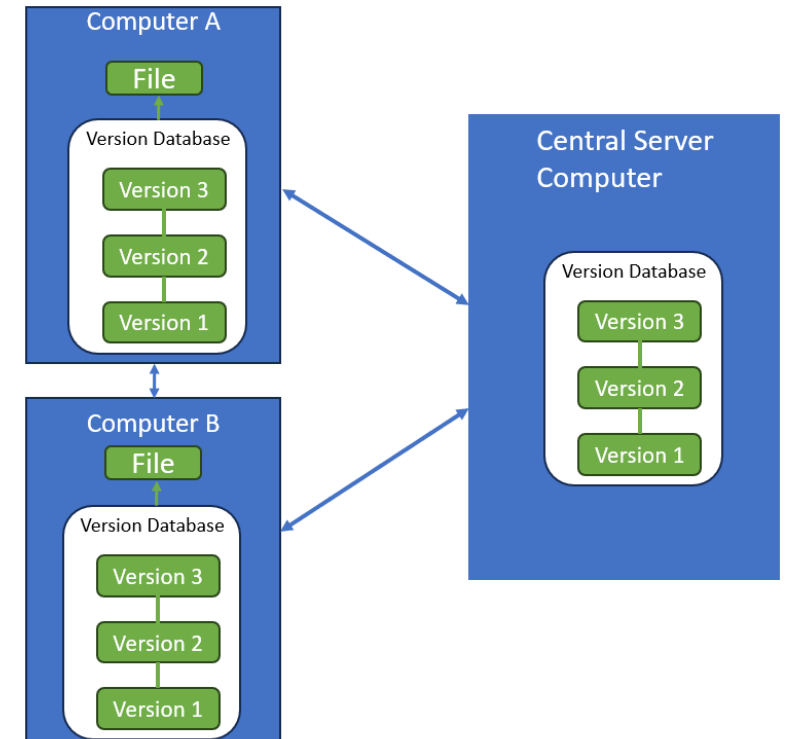


Introduction to Git

- Open-source version control system
- Fast and Distributed
- Easy to learn and team collaboration



Distributed Version Control

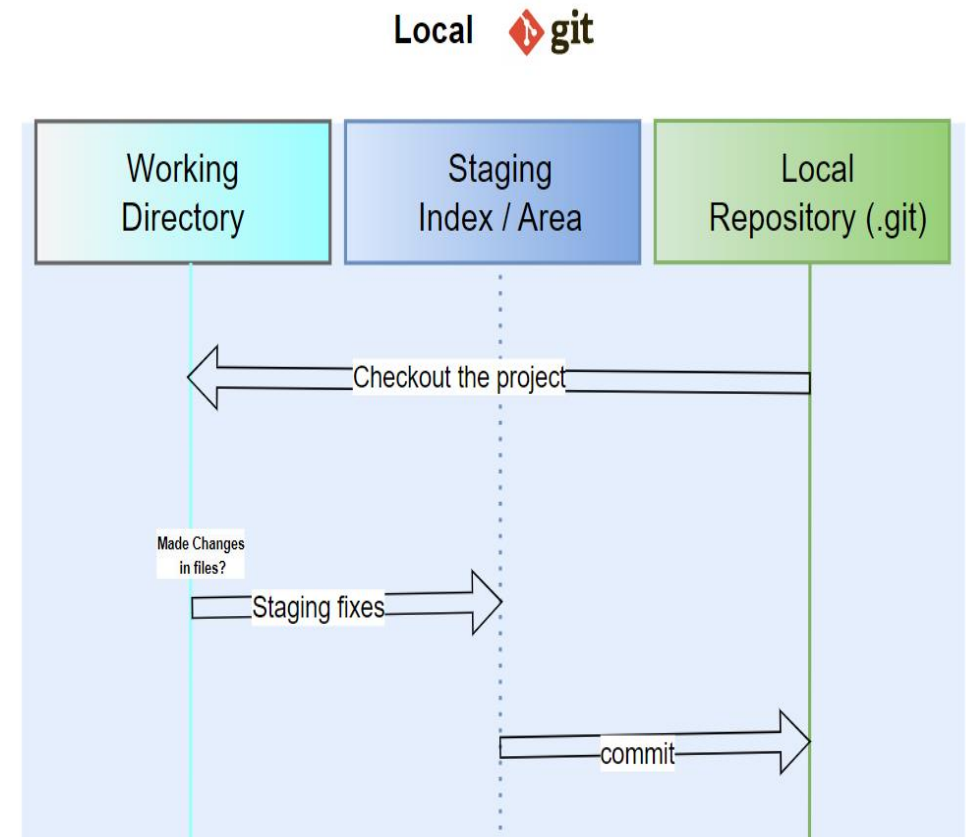


- Git has three states:

- Modified: File has changed but not committed to the database
- Staged: Marked a modified file in its current version. Ready to go in the next commit snapshot.
- Committed: Data is safely stored in the local database.

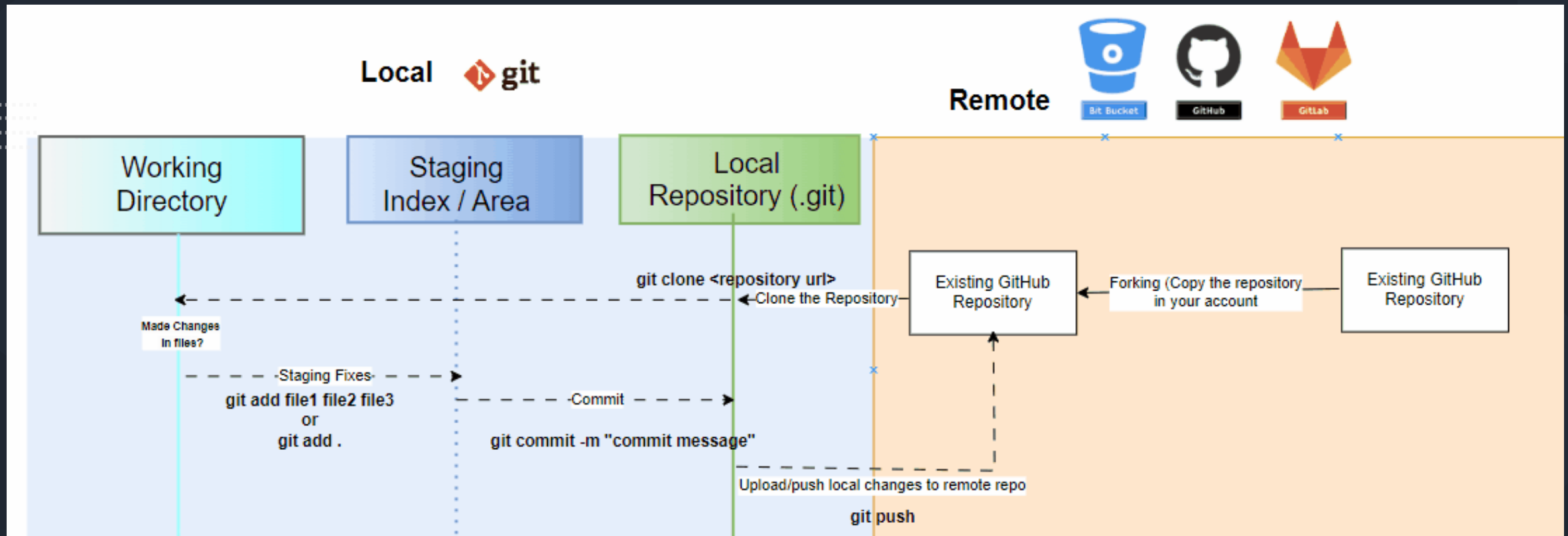
- Git project sections:

- **Working Directory:** Local directory of our project
- **Staging area/index:** Stores information on what will go into the next commit.
- **Git directory:** '.git' directory where Git stores all the information. **If we delete this folder, git will lose all the tracking.**




Git Flow of the Workshop

- GitHub
Repository: <https://github.com/seenaimul/csi-quizzapp>



Download git from: <https://git-scm.com/downloads>

 **git** --local-branching-on-the-cheap

About

Documentation


Downloads


GUI Clients


Logos


Community

Downloads

 **macOS**

 **Windows**

 **Linux/Unix**



Latest source Release
2.43.0
[Release Notes \(2023-11-20\)](#)
[Download for Windows](#)

Older releases are available and the Git source repository is on GitHub.

GUI Clients

Git comes with built-in GUI tools (**git-gui**, **gitk**), but there are several third-party tools for users looking for a platform-specific experience.

The entire **Pro Git book** written by Scott Chacon and Ben Straub is available to [read online for free](#). Dead tree versions are available on [Amazon.com](#).

Create GitHub Account: <https://github.com/>

← → ↻ 📄 github.com ☆ 8 📷 | : 8

All Bookmarks

Product ▾ Solutions ▾ Open Source ▾ Pricing

🔍 Search or jump to... / Sign in Sign up

Let's build from here

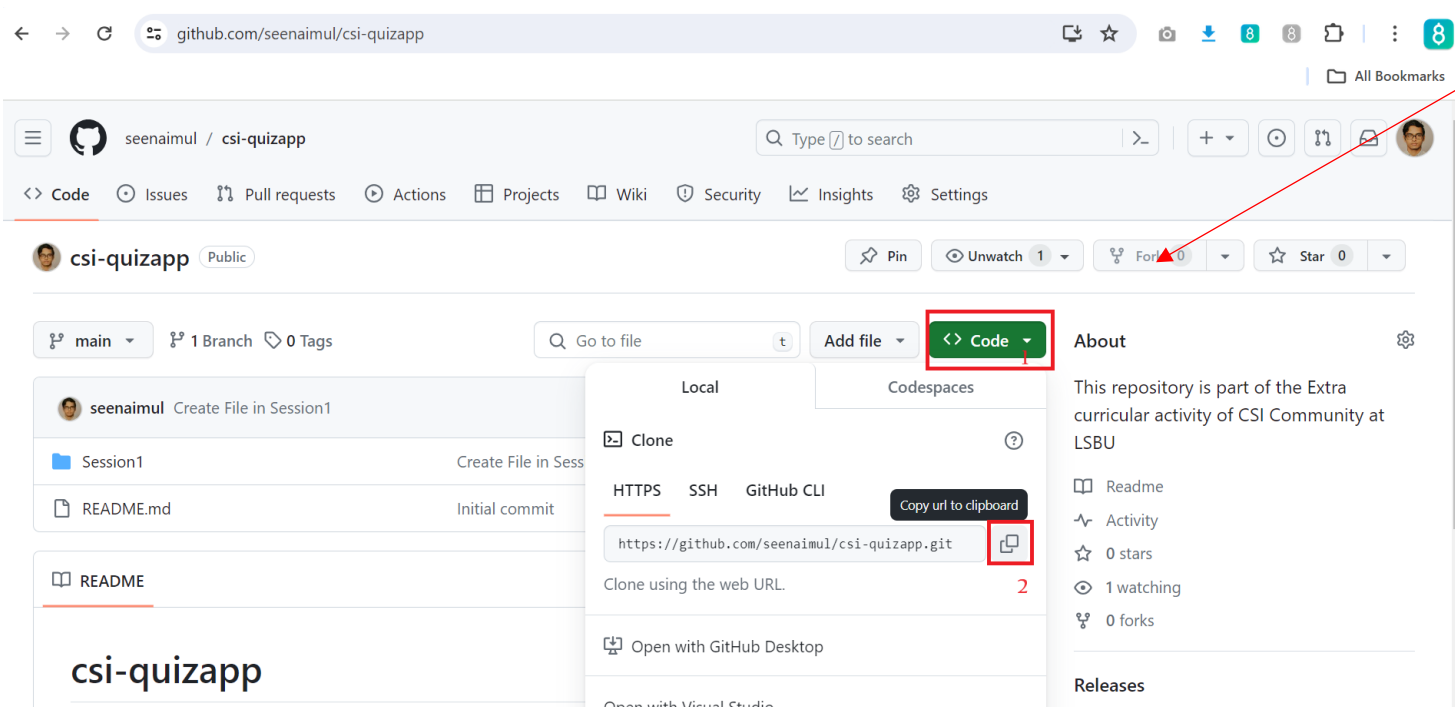
The world's leading AI-powered developer platform.

<>

[Sign up for GitHub](#)

[Start a free enterprise trial >](#)

Copy the GitHub Repo in Your Account



Click on the 'Fork' of the Repo to create a copy in your account.

Create a new fork

A *fork* is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project.

Required fields are marked with an asterisk (*).

Owner *
Repository name *
uniassist1 / csi-quizzapp
csi-quizzapp is available.

By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.

Description (optional)

This repository is part of the Extra curricular activity of CSI Community at LSBU

☒ Copy the main branch only

Contribute back to seenaimul/csi-quizzapp by adding your own branch. [Learn more.](#)


You are creating a fork in your personal account.

Create fork


To clone from any public repository:

- Go inside the repository and Click on the Green button 'Code'
- Copy the HTTPS url of the repository
- Once Git is installed, open any terminal/cmd prompt,
 - Go the location to save the project
 - Type the command 'git clone <url>'
 - For example: **git clone https://github.com/seenaimul/csi-quizzapp.git**

Copy the GitHub Repo in Your Account

 uniassist1 / csi-quizzapp

[code](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)


 **csi-quizzapp** PublicPin Watch 0

forked from [seenaimul/csi-quizzapp](#)

main 1 Branch 0 Tags Add file Code

This branch is up to date with [seenaimul/csi-quizzapp:main](#).

Contribute Sync fork

 **seenaimul** Create File in Session1 3cd2efd · yesterday 2 Commits

Session1	Create File in Session1	yesterday
README.md	Initial commit	2 days ago

README

csi-quizzapp

This repository is part of the Extra curricular activity of CSI Community at LSBU

Clone the Repo from Your account

The screenshot shows the Visual Studio Code interface with the Terminal panel open. The 'Terminal' menu is open, showing options like 'New Terminal' (Ctrl+Shift+') and 'Split Terminal' (Ctrl+Shift+5). The terminal output shows the following commands and their results:

```
naimu@LAPTOP-HBOBMRD1 MINGW64 ~  
$ cd "C:\Users\naimu\OneDrive - London South Bank University\Desktop" 4. Going inside the Desktop Directory  
  
naimu@LAPTOP-HBOBMRD1 MINGW64 ~/OneDrive - London South Bank University/Desktop  
$ git clone https://github.com/seenaimul/csi-quizzapp.git 5. Using Git Clone to copy the repository locally  
Cloning into 'csi-quizzapp'...  
remote: Enumerating objects: 7, done.  
remote: Counting objects: 100% (7/7), done.  
remote: Compressing objects: 100% (4/4), done.  
remote: Total 7 (delta 0), reused 4 (delta 0), pack-reused 0  
Receiving objects: 100% (7/7), done.  
  
naimu@LAPTOP-HBOBMRD1 MINGW64 ~/OneDrive - London South Bank University/Desktop  
$ cd csi  
CSI_Community/ csi-quizzapp/  
  
naimu@LAPTOP-HBOBMRD1 MINGW64 ~/OneDrive - London South Bank University/Desktop  
$ cd csi-quizzapp 6. Checking the repository folder  
  
naimu@LAPTOP-HBOBMRD1 MINGW64 ~/OneDrive - London South Bank University/Desktop/csi-quizzapp (main)  
$ ls -a  
./ ../ .git/ README.md Session1/ 7. All files, Folders are inside along with the hidden '.git' folder.  
  
naimu@LAPTOP-HBOBMRD1 MINGW64 ~/OneDrive - London South Bank University/Desktop/csi-quizzapp (main)
```

Annotations in the image include:

- 1. Arrow pointing to the 'Terminal' menu.
- 2. Arrow pointing to the 'New Terminal' option.
- 3. Arrow pointing to the shell dropdown menu (showing 'bash' and a '+' icon).
- 4. Arrow pointing to the directory path in the terminal command.
- 5. Arrow pointing to the repository URL in the terminal command.
- 6. Arrow pointing to the repository name in the terminal command.
- 7. Arrow pointing to the '.git' folder in the terminal output.

First Commit and Global Configurations

```
File Edit Selection View Go ... < > csi-quizzapp
```

EXPLORER Click to add a file 'index.html'

Session1

index.html

test.txt

README.md

Git can see there is one modification made.

OUTLINE

TIMELINE

EV3DEV DEVICE BROWSER

RUNME NOTEBOOKS

Session1 > index.html

1

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

bash

```
naimu@LAPTOP-HBOBMRD1 MINGW64 ~/OneDrive - London South Bank University/Desktop/csi-quizzapp (main)
$ git status Checking the status of the file system
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  Session1/index.html

nothing added to commit but untracked files present (use "git add" to track)

naimu@LAPTOP-HBOBMRD1 MINGW64 ~/OneDrive - London South Bank University/Desktop/csi-quizzapp (main)
$ git add . Add all the changes in the current directory to the
Staging index/area

naimu@LAPTOP-HBOBMRD1 MINGW64 ~/OneDrive - London South Bank University/Desktop/csi-quizzapp (main)
$ git commit -m "Added index.html file" Making the commit. -m means message
[main 708a49e] Added index.html file
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Session1/index.html

naimu@LAPTOP-HBOBMRD1 MINGW64 ~/OneDrive - London South Bank University/Desktop/csi-quizzapp (main)
$ git push Push all the changes from Git to GitHub repository
Enumerating objects: 6, done.
```

main* 0 0 0 Git Graph Ln 1, Col 1 Spaces: 4 UTF-8 CRLF HTML Go Live

First Commit and Global Configurations

```
naimu@DESKTOP-KTKLFPH MINGW64 ~/OneDrive/Desktop/csi-quizzapp (main)
$ git add . && git commit -m "Folder created"
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

naimu@DESKTOP-KTKLFPH MINGW64 ~/OneDrive/Desktop/csi-quizzapp (main)
$ git push
```

In the first ever Git push, GitHub will show the prompt for authentication.

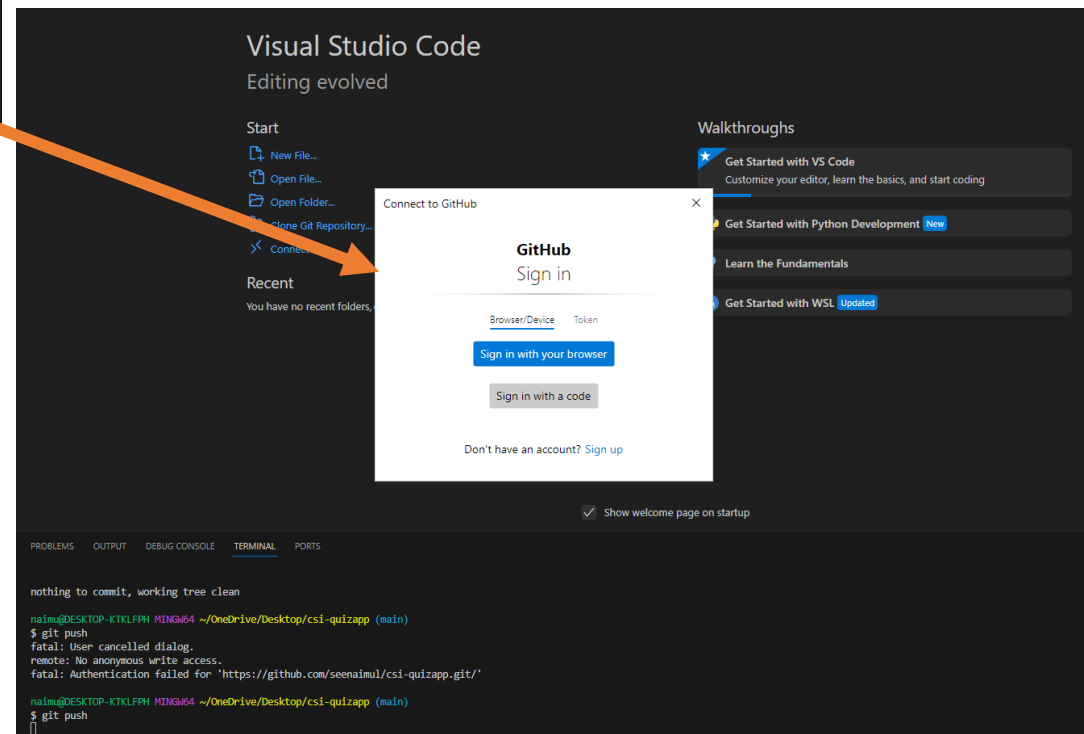
```
naimu@DESKTOP-KTKLFPH MINGW64 ~/OneDrive/Desktop/csi-quizzapp (main)
$ git config --global user.email "naimulmuhammad@gmail.com"

naimu@DESKTOP-KTKLFPH MINGW64 ~/OneDrive/Desktop/csi-quizzapp (main)
$ git config --global user.name "seenaimul"
```

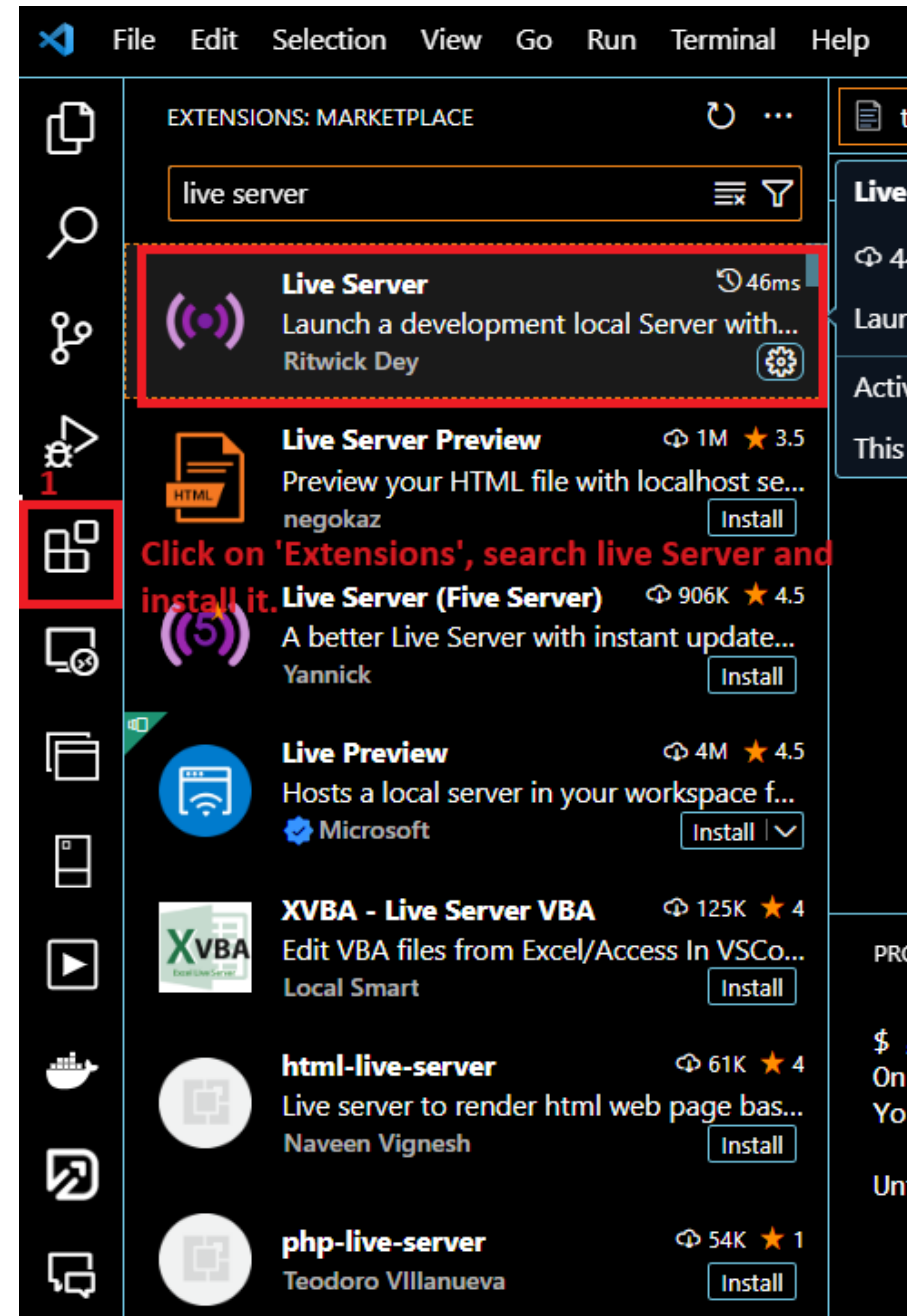
When using for the first time, we should set these global configuration of user name and email.

'git config --list' will list all the configurations

Note: keeping the same username and email of the GitHub account is my personal choice.



Installation of Live Server Extension





What is web development?

- Web Development usually refers to developing the website for the Internet (World Wide Web) or for an Intranet (Private network).
- Also known as Web Programming.
- It is the creation of Dynamic Web Applications.
- Examples of Web Applications are Social networking sites like Facebook or E-commerce sites like Amazon, Flipkart, etc.
- There are two broad division of Web Development.

Front-end Development (also called Client-side Development)

Back-end Development (also called Server-side Development).

Front-End Development

- Front end development refers to producing a web application so that a user can see and interact with them directly.
- It is also known as Client-side development.
- It focuses on the visual elements of a website that a user will interact with.
- A front-end developer has one general responsibility: to ensure that website visitors can easily interact with the page. They do this through the combination of design, technology and programming to code a website's appearance, as well as taking care of debugging.
- The common technologies we can use in front-end development are:

HTML

CSS

JAVASCRIPT

The foundations of front-end development are...



HTML (Hyper Text Markup Language)

- Hypertext means that the document contains links that allow the reader to jump to other places in the document.
- A Markup Language is a way that computers speak to each other to control how text is processed and presented.
- It describes the structure of a web page and consists series of elements. It acts as a skeleton for a website since it is used to make the structure of a website.
- Its elements tells browser how to display the content.
- Its code is written in Notepad or any text editor but save it as .htm or .html extension.



Structure of an HTML program

```
<!DOCTYPE html> ← Tells the document type
<html> ← The Root Element
  <head> ← Contains the header information
    <title>Title of the Page</title> ← Defines Title of
                                     the Page
  </head>
  <body> ← Holds the Content of the Page
    Tags related to layout and formatting
  </body>
</html>
```

Common HTML Tags

Tag	Tag Name	Definition
<code><p> </p></code>	Paragraph	Define the start and stop of a paragraph.
<code><h1> </h1></code>	Heading 1	Headings can range from 1 to 6. The higher the number, the smaller the heading.
<code>
</code>	Line Break	Inserts a single line break. Similar to hitting the carriage return key.
<code><div> </div></code>	Division	A division is a container that holds other elements. A division is helpful when separating blocks of HTML.
<code> </code>	Italic text	<i>Italic Text</i> , a.k.a., 'emphasized text'

``



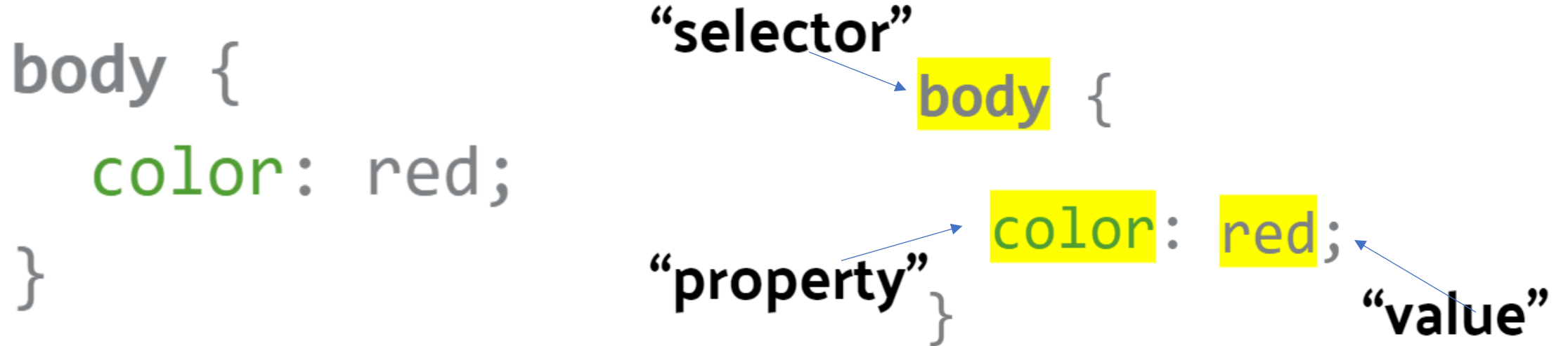
CSS (Cascading Style Sheets)

- Cascading Style Sheets (CSS) describes how HTML elements are to be displayed on screen. It is a simply designed language intended to simplify the process of making web pages presentable. It is used to style our website.
- CSS saves a lot of work. It can control the layout of multiple web pages all at once.
- It is also responsible for responsive layouts of a website.
- It is also written in any text editor but save as .css extension.



Let's look at the syntax

- For example: text should be red.



- The **selector** is `body`, which targets all text within the `<body>` element.
- The **property** is `color`, specifying the text color.
- The **value** assigned to the property is `red`.

What do these properties/values do?

background-color:
lightblue;

cursor: pointer;

display: none;

font-weight: 700;

text-align:
center;

background-image:
url("exampleImageName.jpg");

text-decoration: line-through;

- background-color: light-blue;: This property sets the background color of an element to light blue.
- cursor: pointer;: When applied to an element, it changes the cursor to a pointer (usually a hand) when hovering over that element. This indicates that the element is clickable.
- display: none;: This property hides an element from view. It effectively removes the element from the layout, making it invisible.
- font-weight: 700;: This property makes the text within an element bold. The value 700 corresponds to the bold font weight.
- text-align center;: When applied to a block-level element (like a <div>), it horizontally centers the text content within that element.
- background-image: url("exampleImageName.jpg");: This property sets a background image for an element. The specified image file (in this case, "exampleImageName.jpg") will be used as the background.
- text-decoration: line-through;: When applied to text, it adds a strike-through line over the text. This is commonly used to indicate deleted or irrelevant content.

JavaScript

- JavaScript is a scripting language used to provide a dynamic behavior to our website.
- JavaScript is the Programming Language for the Web. JavaScript can update and change both HTML and CSS.
- JavaScript Allows web pages to be interactive Web pages can change as the user moves the mouse, clicks or types.
- JavaScript is responsible for the Functioning of the website.
- It is a scripting language that enables us to create dynamically updating content, control multimedia, animate images, and pretty much everything else.
- It is also written in any text editor but add the .js extension.



3 Layers of a Web Page

1. Content/Structure - **HTML**
2. Presentation - **CSS**
3. Action/Behavioral - **JavaScript**

- Think of it like the **skeleton** of a building. Just as a building's structure determines its layout and rooms, HTML organizes the content on a web page
- Imagine the same building from before, but now it's adorned with **aesthetic enhancements**—like paint, windows, and decorative elements. CSS adds visual appeal to the web page.
- Picture an **escalator** in our building. It moves, responds to people stepping on it, and changes direction. Similarly, JavaScript adds functionality and responsiveness to your site.

EXAMPLE:

Three layers of web design:

Structure

HTML markup



Style

CSS



Behavior

JavaScript



Back-End Development

- Back-end development controls what goes on behind the scenes of the web applications.
- It is also known as server-side development.
- Back-end usually consists of three parts:
 - **A server**
 - **An Application**
 - **A Database**
- Users can't see how the backend works but this code is what communicates the database information to the browser.

Common Back-End development technologies

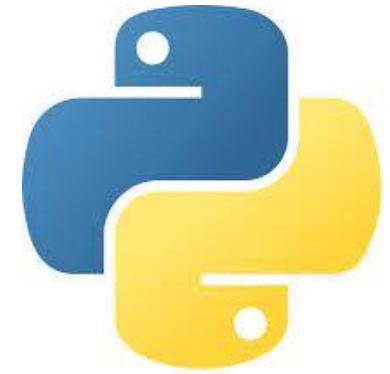
PHP

- **PHP**, which stands for “**PHP: Hypertext Preprocessor**”, is a widely-used, open-source scripting language. It operates on the server side, executing scripts to handle various tasks. Being free to download and use, PHP is a powerful tool for web development. It is embedded within HTML, allowing developers to manage dynamic content, databases, session tracking, and even build entire e-commerce sites.



Python

- Python is a programming language that lets you work quickly and integrate systems more efficiently.



Node.js

- Node.js is an open source and cross- platform runtime environment for executing JavaScript code outside a browser.



Who is a Web Developer?

A web developer is a programmer who specializes in the development of World Wide Web applications using a client-server model.

The applications typically use HTML, CSS and JavaScript in the client, PHP, C#, Python, Node.js or Java in the server, and http for communications between client and server.

There are 3 types of Web Developers -

- Front-End Developer,
 - Back-End Developer,
 - Full Stack Developer.
- **Front-End Developer** :- They are responsible for how a website looks. They create the site's layout and integrate graphics, applications (such as a retail checkout tool), and other content. They also write web design programs in a variety of computer languages, such as HTML or JavaScript.
 - **Back-End Developer** :- They are responsible for server-side web application logic and integration of the work front-end developers do.
 - **Full-Stack Developer** :- A Full-Stack Developer is someone familiar with both front and back-end development. They are generalists, adept at wearing both hats, and familiar with every layer of development. Obviously, employers want to hire Full-Stack Developers - according to an Indeed study, they are the fourth-most in-demand job in tech.

What Does a Web Developer Do?

- **Translating wireframe designs into working code:** Web developers take visual wireframes or design mockups and turn them into functional websites by writing HTML, CSS, and JavaScript code.
- **Creating the architecture and content of a site:** This involves structuring the website, organizing pages, creating navigation menus, and adding textual content.
- **Building in functionality and responsivity:** Web developers implement interactive features, such as forms, buttons, sliders, and responsive layouts that adapt to different screen sizes.
- **Making a site go live:** After development and testing, web developers deploy the site to a live server, making it accessible to users.
- **Updating and renovating sites:** Maintenance is crucial. Developers continuously update content, fix bugs, and improve performance.
- **Troubleshooting, fixing bugs, and glitches:** When issues arise, web developers diagnose problems and apply solutions to ensure smooth functionality.
- **Monitoring website traffic:** Developers track user behavior, analyze metrics, and optimize the site for better performance.

Conclusion

In today's Web development, a good page design is essential. A bad design will lead to the loss of visitors and that can lead to a loss of business. In general, a good page layout must satisfy the basic elements of a good page design.

This includes color contrast, text organization, font selection, style of a page, page size, graphics used, and consistency. In order to create a well-designed page for a specific audience.