ITE 1008 – OSP

DIGITAL ASSIGNMENT 1

CASE STUDY ON GITHUB VERSION CONTROL

NAME : NIBRAS IBRAHIM MOHAMMED ALI

REG NO. : 19BIT0351

Working methodology and was to access github

Step 1: Understanding github

GitHub is a highly used software that is typically used for version control. It is helpful when more than just one person is working on a project. Say for example, a software developer team wants to build a website and everyone has to update their codes simultaneously while working on the project. In this case, Github helps them to build a centralized repository where everyone can upload, edit, and manage the code files.

Step 2: How to create a GitHub Repository?

A repository is a storage space where your project lives. It can be local to a folder on your computer, or it can be a storage space on GitHub or another online host. You can keep code files, text files, images or any kind of a file in a repository. You need a GitHub repository when you have done some changes and are ready to be uploaded. This GitHub repository acts as your remote repository.

Go to the link: https://github.com/ . Fill the sign up form and click on “Sign up for Github”.

Click on “Start a new project”.

Step 3: Create Branches and Perform Operations

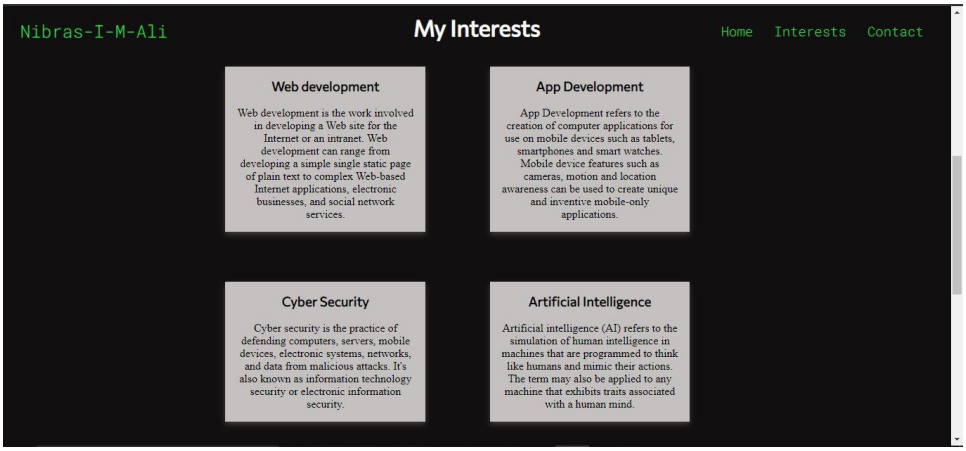
Branching: Branches help you to work on different versions of a repository at one time. Let’s say you want to add a new feature (which is in the development phase), and you are afraid at the same time whether to make changes to your main project or not. This is where git branching comes to rescue. Branches allow you to move back and forth between the different states/versions of a project. In the above scenario, you can create a new branch and test the new feature without affecting the main branch.

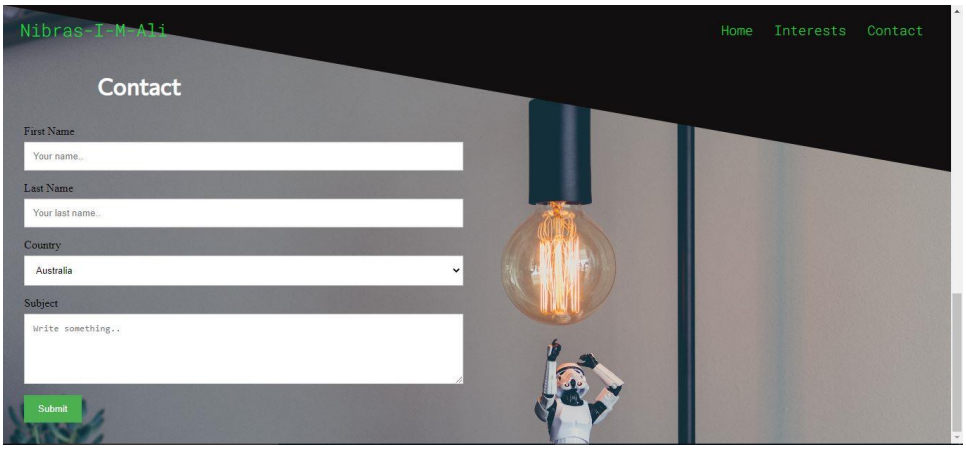
Personal portfolio website

Website link : https://nibras-i-m-ali.github.io/Nibras-19BIT0351/

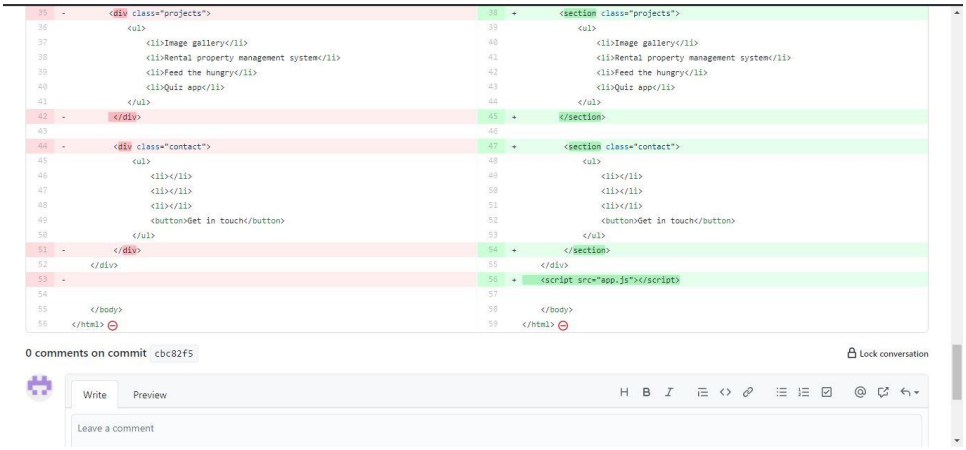
Repo link : https://github.com/Nibras-I-M-Ali/Nibras-19BIT0351

Screenshots





Version history



5 assets/js/app.js

```
@@ -1,3 +1,4 @@
+ console.log("Hello");
  const texts = ["Hello world!", "Nibras here"];
  let count = 0;
  let index = 0;
@@ -12,12 +13,12 @@ let currentText = "";
      currentText = texts[count];
      letter = currentText.slice(0, ++index);

-     document.querySelector(".main_plate").textContent = letter;
+     document.querySelector(".typing").textContent = letter;
      if(letter.length === currentText.length){
          count++;
          index = 0;
        }
-     setTimeout(type, 400);
+     setTimeout(type, 300);
      }
  )();
```

14 assets/main.css

```
@@ -53,16 +53,20 @@ body{

  .main_plate{
      padding-top: 250px;
+ }
+
+ .typing{
      position: relative;
  }

- .main_plate::after{
+ .typing::after{
      content: '';
      position: absolute;
-     border-right: 5px solid  rgb(30, 233, 57);
-     height: 7%;
-     width: 1px;
-     animation: blink 0.5s infinte easy;
+     border-right: 5px solid rgb(30, 233, 57);
+     height: 90%;
+     right: -7px;
+     width: 2px;
+     animation: blink 0.5s infinite ease;
  }

  @keyframes blink{
```

9 index.html

```
@@ -13,13 +13,14 @@

          <section class="main_banner">
              <div class="main_plate">
-                 Hello World!
-                 Nibras here
+                 .
+                 <span class="typing"></span>
              </div>

              <div class="desc">
                  A tech enthusiast student at VIT
                  <br>
-                 scroll to get to know us
+                 scroll to get to know me
              </div>
          </section>

@@ -53,7 +54,7 @@
              </ul>
          </section>
      </div>
-     <script src="app.js"></script>
+     <script type="text/javascript" src="assets/js/app.js"></script>

      </body>
  </html>
```

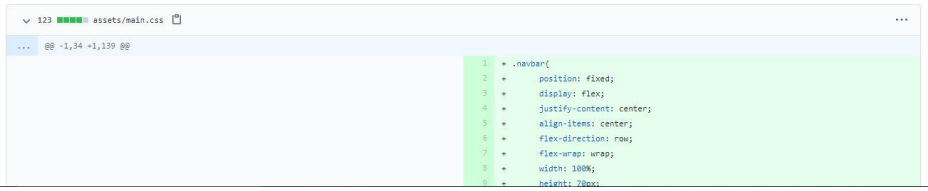
0 comments on commit `0b13e07`

Lock conversation

5 assets/js/app.js

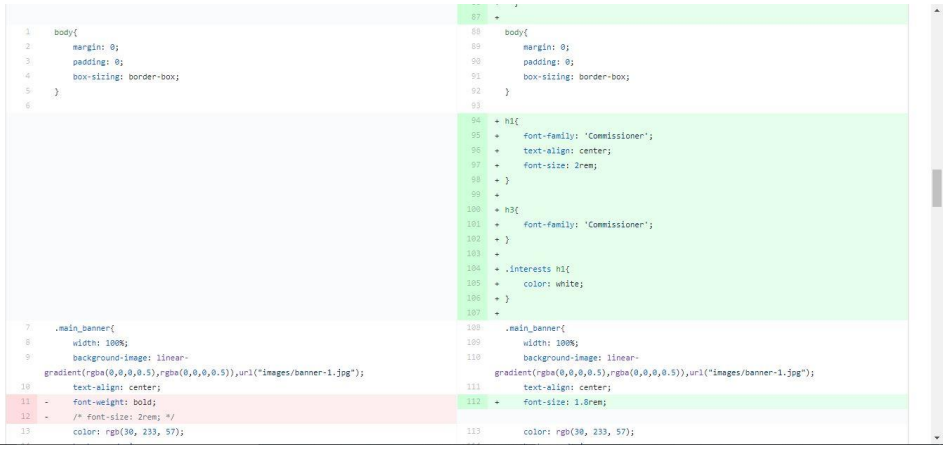
```
@@ -1,10 +1,11 @@
- console.log("Hello");
- const texts = ["Hello world!", "Nibras here"];
+ const texts = ["Hello world! Nibras here"];
+ const texts1 = [""]
  let count = 0;
  let index = 0;
  let currentText = "";


+
  (function type()
    {
      if(count === texts.length){
```



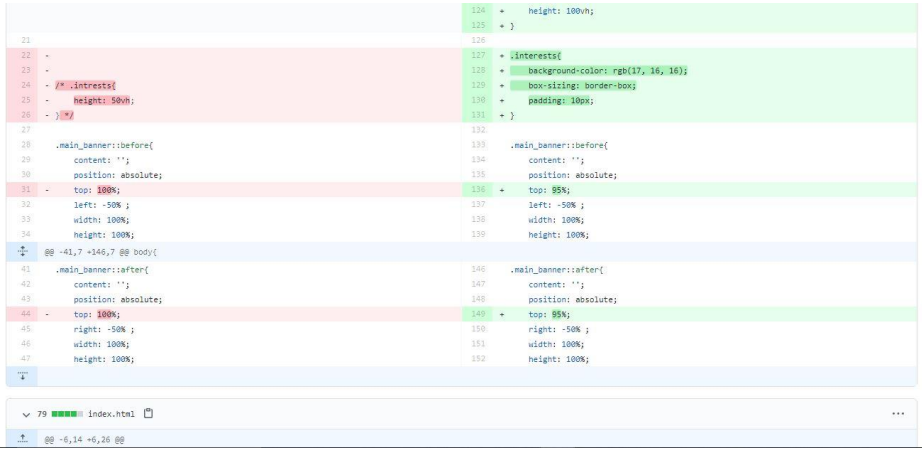
123 assets/main.css

```
@@ -1,34 +1,139 @@
+ .navbar{
+       position: fixed;
+       display: flex;
+       justify-content: center;
+       align-items: center;
+       flex-direction: row;
+       flex-wrap: wrap;
+       width: 100%;
+       height: 70px;
```



```
  body{
      margin: 0;
      padding: 0;
      box-sizing: border-box;
  }

+ h1{
+     font-family: 'Commissioner';
+     text-align: center;
+     font-size: 2rem;
+ }
+
+ h3{
+     font-family: 'Commissioner';
+ }
+
+ .interests h1{
+     color: white;
+ }
+
  .main_banner{
      width: 100%;
      background-image: linear-gradient(rgba(0,0,0,0.5),rgba(0,0,0,0.5)),url("images/banner-1.jpg");
      text-align: center;
-     font-weight: bold;
-     /* font-size: 2rem; */
+     font-size: 1.8rem;
      color: rgb(30, 233, 57);
```



```
+     height: 100vh;
+ }

- /* .intrests{
-     height: 50vh;
- } */
+ .interests{
+     background-color: rgb(17, 16, 16);
+     box-sizing: border-box;
+     padding: 10px;
+ }

  .main_banner::before{
      content: '';
      position: absolute;
-     top: 100%;
+     top: 95%;
      left: -50% ;
      width: 100%;
      height: 100%;
@@ -41,7 +146,7 @@ body{
  .main_banner::after{
      content: '';
      position: absolute;
-     top: 100%;
+     top: 95%;
      right: -50% ;
      width: 100%;
      height: 100%;
```


79 index.html

```
@@ -6,14 +6,26 @@
```

```
-  /* Float four columns side by side */
   .column {
     float: left;
     width: 40%;
     padding: 0 10px;

   }
-  /* Remove extra left and right margins, due to padding */
   .row {margin: 0 -5px}
-  /* Clear floats after the columns */
   .row:after {
     content: "";
     display: table;
     clear: both;
   }
-  /* Responsive columns */
   @media screen and (max-width: 600px) {
     .column {
       width: 100%;
@@ -76,7 +77,6 @@ a{
     }
   }
-  /* Style the counter cards */
```

```
   .column {
     float: left;
     width: 40%;
     padding: 0 10px;

   }
   .row {margin: 0 -5px}
   .row:after {
     content: "";
     display: table;
     clear: both;
   }
   @media screen and (max-width: 600px) {
     .column {
       width: 100%;
     }
   }
```

Pros and cons of github:

Pros:

1. Markdown

Markdown allows you to use a simple text editor to write formatted documents. GitHub, like many online repo services, supports Markdown for the issue tracker, user comments, wikis – everything. With so many other programming languages to learn for setting up projects, it's really a big benefit to have your content inputted in a format without having to learn yet another system. In addition, there is also what is known as the GitHub flavored markdown – a feature that adds changes to the usual markdown in order to make it more useful in programming environments.

2. GitHub has some of the best documentation around

You won't run out of content when you use GitHub, thanks to a well-padded guide and help section for articles that you can pull up for practically any topic on earth, for as long as it is related to a git. It's got content for helping you learn about generating SSH keys. A guide for the best git workflow is available. Samples on gitignore (and more) are abound for your next planned project, among other things. You would not need to look elsewhere for all the information that you need.

3. GitHub has Gists and GitHub Pages, too

A while back, GitHub rolled out a feature called Gists, which lets you convert one or several files into a working git repository. This new feature converted sharing and tracking changes made to configuration files and even simple scripts into a whole new level of easy. While they aren't as rich in features like a full-blown GitHub repository, they really work well even if you are without a paid account. GitHub pages, on the other hand, lets you host static websites by simple

assigning HTML pages onto another, separate repository – the way you would any other type of git repository. With this, blogging can be done off the bat as well as updating with additional documentation or bumping up its web presence.

4. Collaboration

For those who are not in the same physical location, an online Git is an easy solution requiring no setup for new users. With no need to connect to the company's VPN, it may be easier to dump everything on a private repository on GitHub.

This benefit is much greater to those working collaboratively on a project that are not part of a professional environment – particularly open source projects. Most programmers are already familiar with how to use GitHub, and it's easy to point people to a GitHub page if they want to make contributions. Online repositories are essential for open source projects, and the only reason some may avoid GitHub was acquired by Microsoft some time ago, which resulted in many switching to alternative like GitLab. While there's no current reason in particular for this switch, many do not trust Microsoft's track record of attacking the open source community in the past.

5. Backup

Using an online repository should never be considered infallible, but it provides a nice and simple way to have their code and version history available online, regardless of what happens to their local machine. For some people, this is enough, but we stress that a multi solution backup plan is always the best.

Cons:

1. Security

GitHub does offer private repositories, but this isn't necessarily perfect for many. For high value intellectual property, you're putting all of this in the hands of GitHub as well as anyone who has a login, which like many sites has had security breaches before and is targeted constantly. It is often better than nothing, but it's not perfect. In addition, some clients/employers will only allow code on their own secure internal Git as a matter of policy.

2. Pricing

Some of GitHub features, as well as features on other online repositories, are locked behind a SaaS paywall. If you have a large team, this can add up fast. Those who already have a dedicated IT team and their own internal servers are often better off using their own internal git for cost reasons, but for most the cost isn't outrageous.

3. If you're new to GitHub, one of the challenges often talked about is getting to grips with the mental model, which eases with practice and time.
4. It may not be the best tool for capturing the creative process or for recording ideas. A good tool for this particular function would be LayerVault or something similar to it. We'd say Git is very good for tracking code, however it's not the best for tracking design. It can seem a little bit of a grey area when designs are needed to be translated into code or for when you need to export designs to a production setting.

Features that need to be added:

1. Drag and Drop Gist Code
2. Creating a folder via the Web Interface
3. Git URL Shortener
4. File Finder
5. Linking Lines

Comparison between cvs svn and git

While Git and SVN are both enterprise version control systems (VCS) that help with workflow and project management in coding, they do have their differences. The difference between Git and SVN version control systems is that Git is a distributed version control system, whereas SVN is a centralized version control system. Git uses multiple repositories including a centralized repository and server, as well as some local repositories. SVN does not have a centralized repository or server.

What is GIT?

Git is a distributed version control system - which just means that when you do a **git clone (+url of your repository)** what you are actually getting is a complete copy of your entire history of that project. This means all your commits! Woot!

What are the advantages of Git?

Git has a staging area. This just means that if you made 100 new changes to your code, you can break these 100 changes into 10 or 20 or more commits each with their own comments and their own detailed explanation of what just happened! Not only can you stage your commits out to logically display what changes were made, but you can also do patch staging that ask you if you want. You would use patch staging if you and a co-worker are both working on the same file and you only want to commit a particular function that you've worked on. You do a Git patch using "git add -p"

What is SVN?

Subversion (SVN) may be one of the most well known centralized version control systems. In Subversion or SVN, you are checking out a single version of the repository. With SVN, your data is stored on a central server. Having the entire history on your local repository just means that even when you are not connected to the Internet, you can still do commits, diffs, logs, branches, merges, file annotations, etc.

What are the advantages of SVN?

SVN has one central repository – which makes it easier for managers to have more of a top down approach to control, security, permissions, mirrors and dumps. Additionally, many say SVN is easier to use than Git. For example, it is easier to create a new feature. With Git, it takes an extra step to create a new feature. Others say that the way SVN is set up results in greater trunk stability, and having everything on a central server feels more controlled and secure for some.

CVS

CVS, also known as the Concurrent Versioning System, is a free client-server revision control system in the field of software. It was developed in the UNIX operating system environment and is available in both Free Software Foundation and commercial versions.

Features

- Client-server repository model.
- Multiple developers might work on the same project parallelly.
- CVS client will keep the working copy of the file up-to-date and requires manual intervention only when an edit conflict occurs
- Keeps a historical snapshot of the project.
- Anonymous read access.
- ‘Update’ command to keep local copies up to date.
- Can uphold different branches of a project.
- Excludes symbolic links to avoid a security risk.
- Uses delta compression technique for efficient storage.