K. J. Somaiya College of Engineering, Mumbai-77
(Autonomous College Affiliated to University of Mumbai)

Batch: A2 Roll No.: 1911027

Experiment No. 1

Grade: AA / AB / BB / BC / CC / CD /DD

Title: Study of Cloud Computing & Architecture (Github)

Objective: To understand the cloud computing and architecture of SaaS product (Github)

Expected Outcome of Experiment:

CO	Outcome
CO1	Describe fundamental and core concepts of cloud computing

Books/ Journals/ Websites referred:

1. https://en.wikipedia.org/wiki/Cloud_computing
2. https://en.wikipedia.org/wiki/Cloud_computing_architecture
3. https://github.blog/category/engineering/
4. https://github.com/collections/projects-that-power-github
5. https://blog.hubspot.com/website/what-is-github-used-for

Abstract:-

Cloud computing is the on-demand delivery of IT resources over the Internet with pay-as-you-go pricing. Instead of buying, owning, and maintaining physical data centers and servers, you can access technology services, such as computing power, storage, and databases, on an as-needed basis from a cloud provider. Organizations of every type, size, and industry are using the cloud for a wide variety of use cases, such as data backup, disaster recovery, email, virtual desktops, software development and testing, big data analytics, and customer-facing web applications. For example, healthcare companies are using the cloud to develop more personalized treatments for patients. Financial services companies are using the cloud to power real-time fraud detection and prevention.

Related Theory: -

Cloud computing: Simply put, cloud computing is the delivery of computing services—including servers, storage, databases, networking, software, analytics, and intelligence—over the Internet ("the cloud") to offer faster innovation, flexible resources, and economies of scale. You typically pay only for cloud services you use, helping lower your operating costs, run your infrastructure more efficiently and scale as your business needs change.

Characteristics of cloud computing:

1) **On-demand self-services:** The Cloud computing services does not require any human administrators, user themselves are able to provision, monitor and manage computing resources as needed.
2) **Broad network access:** The Computing services are generally provided over standard networks and heterogeneous devices.
3) **Rapid elasticity:** The Computing services should have IT resources that are able to scale out and in quickly and on as needed basis. Whenever the user require services it is provided to him and it is scale out as soon as its requirement gets over.

4) **Resource pooling:** The IT resource (e.g., networks, servers, storage, applications, and services) present are shared across multiple applications and occupant in an uncommitted manner. Multiple clients are provided service from a same physical resource.

5) **Measured service:** The resource utilization is tracked for each application and occupant, it will provide both the user and the resource provider with an account of what has been used. This is done for various reasons like monitoring billing and effective use of resource.

Advantages of cloud computing:

1) **Cost:** Cloud computing eliminates the capital expense of buying hardware and software and setting up and running on-site datacenters—the racks of servers, the round-the-clock electricity for power and cooling, the IT experts for managing the infrastructure. It adds up fast.

2) **Speed:** Most cloud computing services are provided self service and on demand, so even vast amounts of computing resources can be provisioned in minutes, typically with just a few mouse clicks, giving businesses a lot of flexibility and taking the pressure off capacity planning.

3) **Security:** Many cloud providers offer a broad set of policies, technologies and controls that strengthen your security posture overall, helping protect your data, apps and infrastructure from potential threats.

Disadvantages of cloud computing:

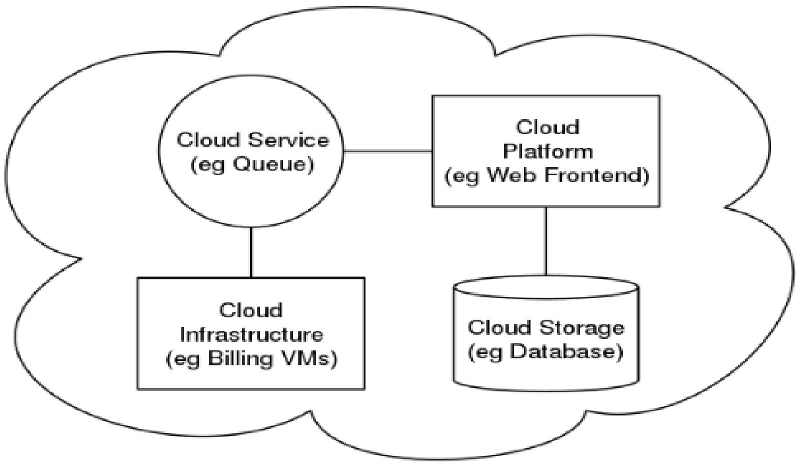
1) **Downtime:** Downtime is often cited as one of the biggest disadvantages of cloud computing. Since cloud computing systems are internet-based, service outages are always an unfortunate possibility and can occur for any reason.

2) **Vulnerability to attack:** In cloud computing, every component is online, which exposes potential vulnerabilities. Even the best teams suffer severe attacks and security breaches from time to time.

3) **Cost concerns:** Adopting cloud solutions on a small scale and for short-term projects can be perceived as being expensive. However, the most significant cloud computing benefit is in terms of IT cost savings.

Cloud computing architecture:

Cloud computing architecture refers to the components and subcomponents required for cloud computing. These components typically consist of a front end platform (fat client, thin client, mobile),back end platforms (servers, storage), a cloud based delivery, and a network (Internet, Intranet, Intercloud). Combined, these components make up cloud computing architecture. A standard internet connection or a virtual network provides us access to cloud-based applications and services like Google Docs, Skype, and Netflix. Most companies are shifting their businesses into the cloud as they require significant storage, which cloud platforms provide.


Cloud Service (eg Queue)
Cloud Platform (eg Web Frontend)
Cloud Infrastructure (eg Billing VMs)
Cloud Storage (eg Database)

A cloud computing architecture provides higher bandwidth to its users due to which data over the cloud can be used from anywhere across the world at any time. Due to its architecture, it not only shares resources among client source consumers but also with open source communities like Microsoft and Red hat.

Types of cloud services:

Most cloud computing services fall into four broad categories: infrastructure as a service (IaaS), platform as a service (PaaS), serverless and software as a service (SaaS). These are sometimes called the cloud computing stack because they build on top of one another. Knowing what they are and how they are different makes it easier to accomplish your business goals.

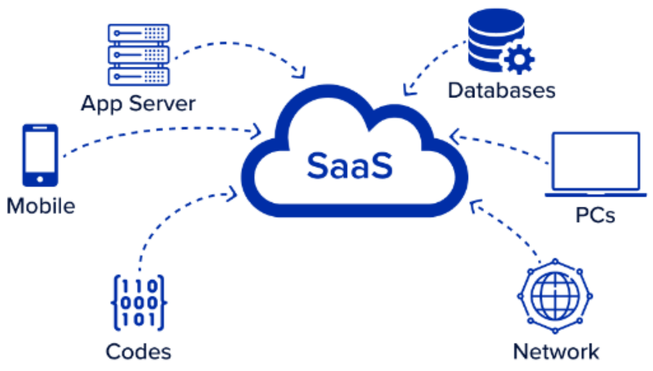
1) **Infrastructure as a service (IaaS):** The most basic category of cloud computing services. With IaaS, you rent IT infrastructure—servers and virtual machines (VMs), storage, networks, operating systems—from a cloud provider on a pay-as-you-go basis.

2) **Platform as a service (PaaS):** Platform as a service refers to cloud computing services that supply an on-demand environment for developing, testing, delivering and managing software applications. PaaS is designed to make it easier for developers to quickly create web or mobile apps, without worrying about setting up or managing the underlying infrastructure of servers, storage, network and databases needed for development.

3) **Software as a service (SaaS):** Software as a service is a method for delivering software applications over the Internet, on demand and typically on a subscription basis. With SaaS, cloud providers host and manage the software application and underlying infrastructure and handle any maintenance, like software upgrades and security patching. Users connect to the application over the Internet, usually with a web browser on their phone, tablet or PC.

SaaS – Software as a service:

SaaS product is an internet software that all users have access to, like a text expander. Many of your favourite internet platforms are SaaS – Google Apps, DropBox, Canva, etc. To use SaaS platforms, you do not need to install anything on your device. A server provider hosts the application. Both the maintenance and updates are performed on the host side and are invisible to the user. The best thing about SaaS software is that you don’t need any special equipment or specialists to use it. Go to the site – use the product – grow profit. Software as a service (or SaaS) is a way of delivering applications over the Internet—as a service. Instead of installing and maintaining software, you simply access it via the Internet, freeing yourself from complex software and hardware management. SaaS applications are sometimes called Web-based software, on-demand software, or hosted software. Whatever the name, SaaS applications run on a SaaS provider’s servers. The provider manages access to the application, including security, availability, and performance.



SaaS characteristics:

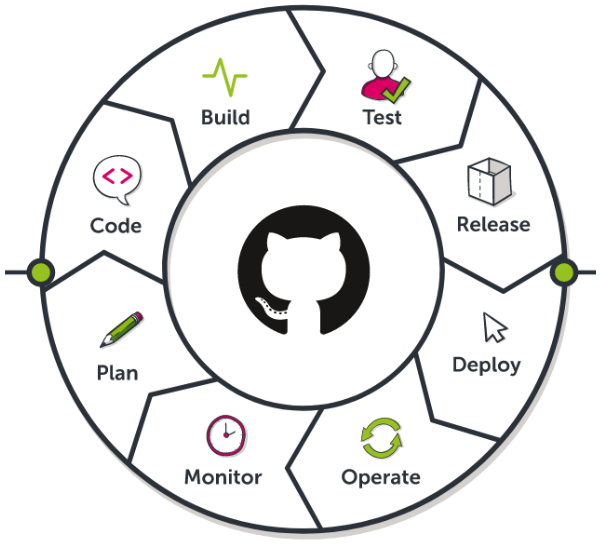
1) **Multitenant Architecture:** A multitenant architecture, in which all users and applications share a single, common infrastructure and code base that is centrally maintained. Because SaaS vendor clients are all on the same infrastructure and code base, vendors can innovate more quickly and save the valuable development time previously spent on maintaining numerous versions of outdated code.

2) **Easy Customisation:** The ability for each user to easily customise applications to fit their business processes without affecting the common infrastructure. Because of the way SaaS is architected, these customisations are unique to each company or user and are always preserved through upgrades.

3) **Better Access:** Improved access to data from any networked device while making it easier to manage privileges, monitor data use, and ensure everyone sees the same information at the same time.

Github (SaaS product):

GitHub, Inc., is an Internet hosting service for software development and version control using Git. It provides the distributed version control of Git plus access control, bug tracking, software feature requests, task management, continuous integration, and wikis for every project. It is commonly used to host open source software development projects. As of June 2022, GitHub reported having over 83 million developers and more than 200 million repositories, including at least 28 million public repositories. Projects on GitHub.com can be accessed and managed using the standard Git command-line interface; all standard Git commands work with it. GitHub.com also allows users to browse public repositories on the site. Multiple desktop clients and Git plugins are also available. The site provides social networking-like functions such as feeds, followers, wikis (using wiki software called Gollum) and a social network graph to display how developers work on their versions ("forks") of a repository and what fork (and branch within that fork) is newest. Anyone can browse and download public repositories but only registered users can contribute content to repositories. With a registered user account, users are able to have discussions, manage repositories, submit contributions to others' repositories, and review changes to code. The fundamental software that underpins GitHub is Git itself. The additional software that provides the GitHub user interface was written using Ruby on Rails and Erlang by GitHub.



Features of github:

1. Easy Project Management: GitHub is a place where project managers and developers come together to coordinate, track, and update their work so that projects are transparent and stay on schedule.

2. Increased Safety With Packages: Packages can be published privately, within the team, or publicly to the open-source community. The packages can be used or reused by downloading them from GitHub.

3. Effective Team Management: GitHub helps all the team members stay on the same page and organized. Moderation tools like Issue and Pull Request Locking help the team to focus on the code.

4. Improved Code Writing: Pull requests help the organizations to review, develop, and propose new code. Team members can discuss any implementations and proposals through these before changing the source code.

5. Increased Code Safety: GitHub uses dedicated tools to identify and analyze vulnerabilities to the code that other tools tend to miss. Development teams everywhere work together to secure the software supply chain, from start to finish.

6. Easy Code Hosting: All the code and documentation are in one place. There are millions of repositories on GitHub, and each repository has its own tools to help you host and release code.

What is GitHub used for?

GitHub allows software developers and engineers to create remote, public-facing repositories on the cloud for free. Once you've set up a repository on GitHub, you can copy it to your device, add and modify files locally, then "push" your changes back to the repository where your changes are displayed to the public.

a) **Enhanced Collaboration:** GitHub provides a centralized space where several, dozens, or even thousands of developers can seamlessly contribute to a project, without worrying about overriding anyone's work or losing track of changes.

b) **Easy File Management:** GitHub adds a sleek graphical user interface (GUI) layer on top of Git. On its own, Git operates through the command line (a computer's text-based interface). Developers know how to use the command line, but for many, it's not always the most efficient way to interact with files. GitHub's interface provides a clean and user-friendly means to perform Git actions as well as view file history. This is more convenient for developers and more accessible for beginners getting the hang of Git.

c) **Open-Source Projects:** GitHub has fueled a surge of open-source collaboration, leading to the creation of many widely used software technologies. From CSS frameworks to data visualization libraries to a game you might spend too much time playing, a lot of impressive feats wouldn't be around without open GitHub repositories.

d) **Social Networking:** Any GitHub user knows the platform is more than just a place to work on code. All GitHub users have profiles to display their projects, contributions, and activity on the site, and can see anyone's public-facing profile and repositories.

Implementation Details:

1. Enlist all the Steps followed and various options explored

➔ GitHub's architecture is deeply rooted in Ruby on Rails.

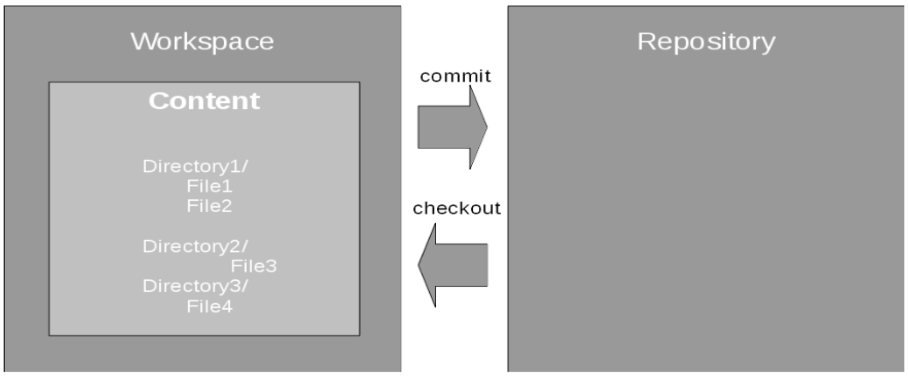
Internal working of github:

a) Git uses the SHA-1 hash of content to create references to commits, trees and blobs.
b) A commit object stores the metadata about a commit, such as the parent, the author, timestamps and references to the file tree of this commit.
c) A tree object is a collection of references to either child trees or blob objects.

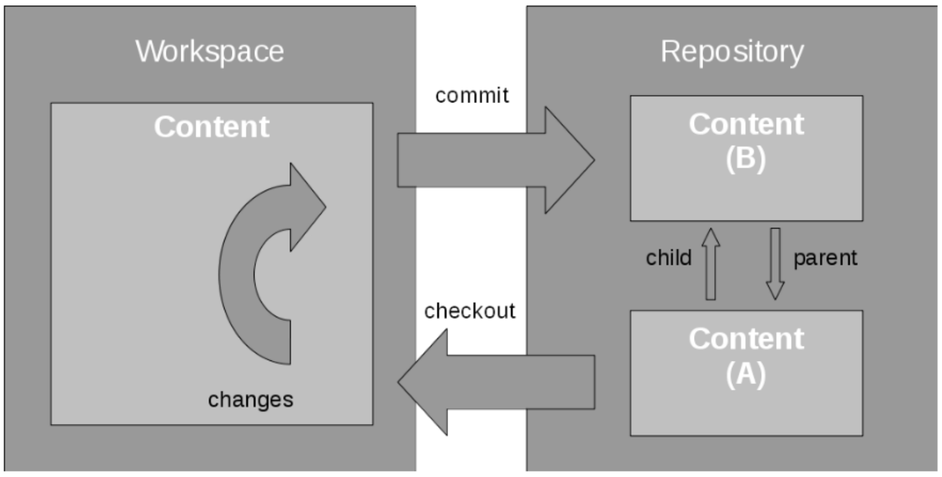
d) Blob objects are compressed collections of files; usually, the set of files in a particular directory inside the tree.

Steps:

1) Typical source code repository, folder with files and subfolders are handled as the content (CVS and Git don't actually handle folders, just files at a path location). The repository holds all versions of the content, while the working directory is the place where you modify the code.

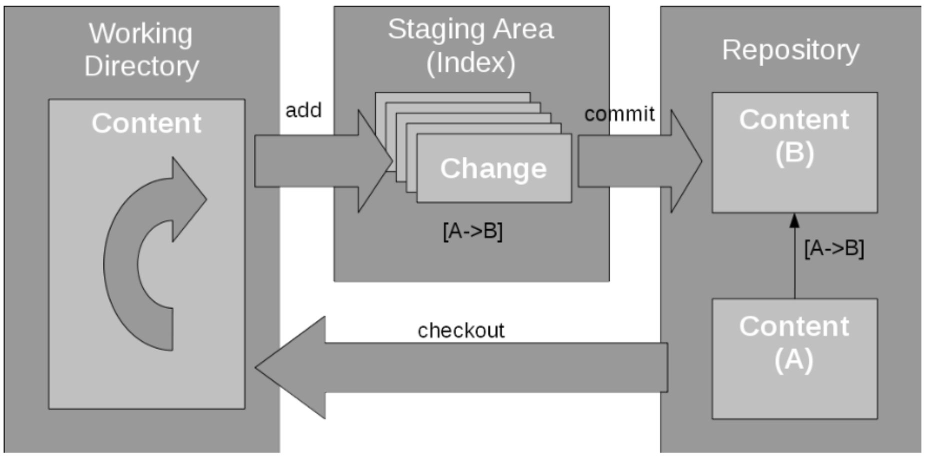


2) Each commit creates a new child version of the content that derives from the previous, parent version that you modified. Each commit creates a new child version of the content that derives from the previous, parent version that you modified. This series of versions is called a stream or branch. In SVN, the main stream is called trunk; in CVS it usually goes by the name HEAD; in Git it is usually named master.

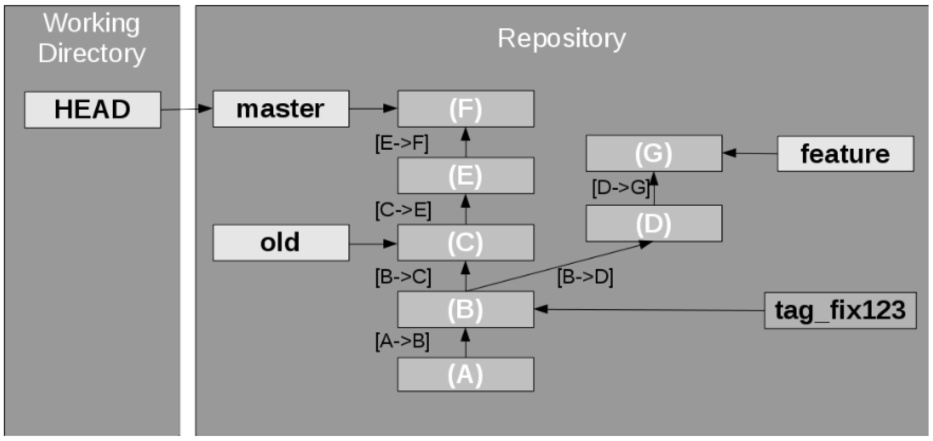


3) **git init** - Initializes a repository
4) **git checkout <branch>** - Checks out a branch from repository into the working directory
5) **git add <file>** - Adds a change in a file to a change set
6) **git commit** - Commits a change set from the working directory into the repository

7) **git status** - Helps you keep track of which changes have been added, or not, on what branch you are on.



8) **git log** - Shows the history of the changes (i.e. commits) in the working directory, or with git log <path> the changes applied to the given path.
9) The concept behind branching is that each snapshot can have more than one child. Applying a second change set to the same snapshot creates a new, separate stream of development. And if it is named, it is called a branch.



Various options explored:

1) **Bitbucket:** Bitbucket is version control tool developed by Atlassian. It is more than just Git code management. Bitbucket is one of the best alternatives to GitHub which allows the team to plan projects, collaborate on code, test, and deploy.
Features: a) Free unlimited private repositories.
b) Best-in-class Trello & Jira integration.
c) Allows you to build quality software with code review.

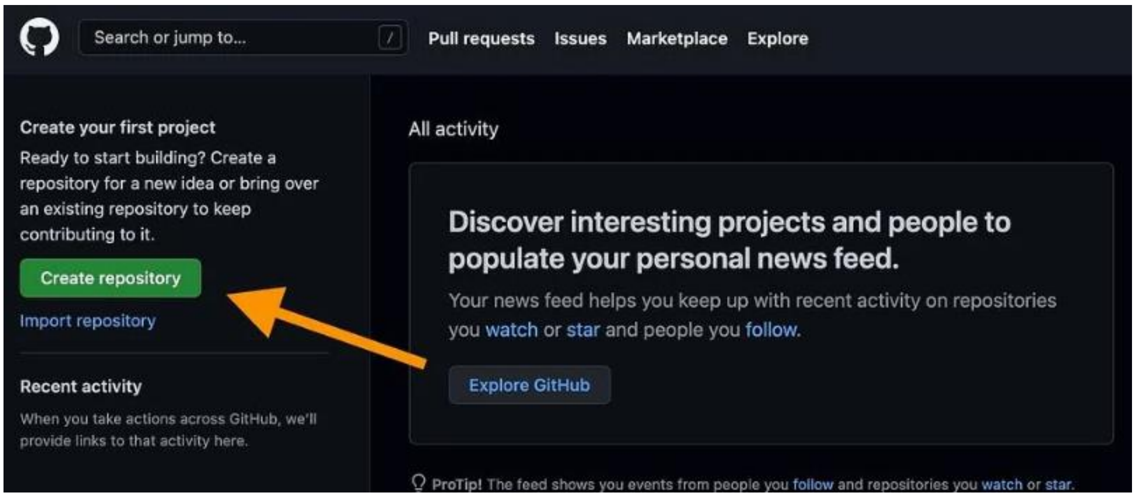
2) **SourceForge:** SourceForge is an open source development and distribution platform. The tool is hosted on Apache, Allura, and supports many different projects. Users can select either Git, Mercurial as their version control system.
   Features: a) Extensive worldwide mirror network.
   b) Integrated Issue Tracking.
   c) This GitHub alternative open source allows browser-based code browsing.
3) **TaraVault:** TaraVault is Inflectra's free cloud-based source code management solution for enterprises and teams of all sizes.
   Features: a) The ability to link requirements, tasks, defects, and issues to source code files and revisions for maximum traceability.
   b) Integrated ALM and issue-tracking for your projects.
   c) Source code browsing, inline code diffs, and pull request management.

2. Explain your program logic, classes and methods used, as applicable.

➔ Git uses the terminal as its interface. Github also provides graphical user interface platform from where a user can make changes to github repositories from desktop itself.

Basic overview of how to use github:

1) Install the latest version of Git on your device. You'll need Git installed to work with your GitHub repository.
2) After installing Git, go to GitHub's website and create an account with your email address.
3) Once your GitHub account is set up, you'll be taken to your dashboard. To start your first repository, click Create repository on the left side.



Create your first project

Ready to start building? Create a repository for a new idea or bring over an existing repository to keep contributing to it.

Recent activity

When you take actions across GitHub, we'll provide links to that activity here.

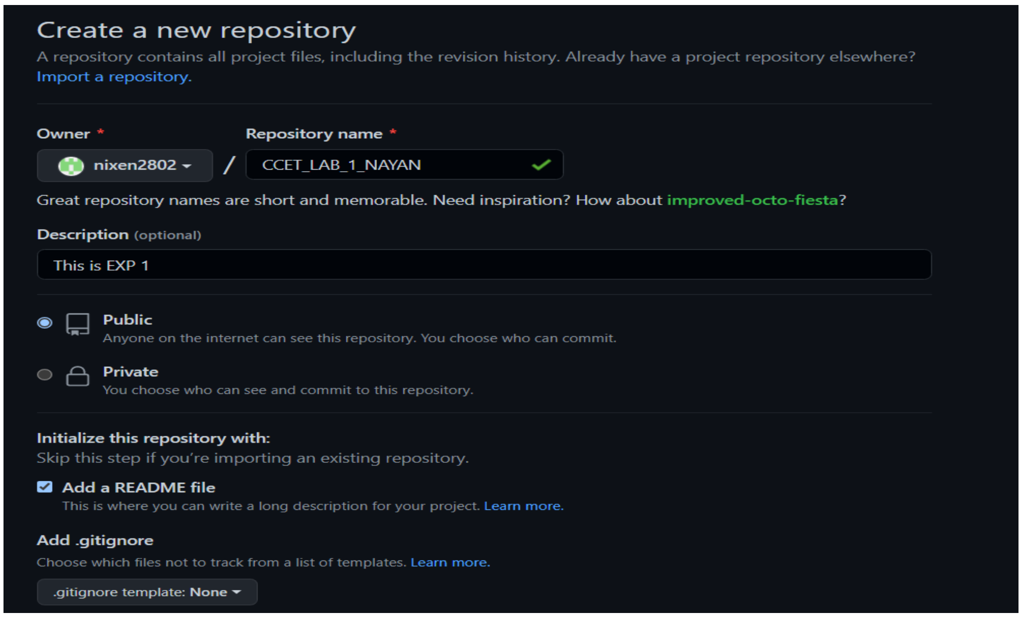
All activity

Discover interesting projects and people to populate your personal news feed.

Your news feed helps you keep up with recent activity on repositories you watch or star and people you follow.

ProTip! The feed shows you events from people you follow and repositories you watch or star.

4) On the Create a new repository screen, enter your repository name and an optional description (you can change both later).



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner * nixen2802 / Repository name * CCET_LAB_1_NAYAN

Great repository names are short and memorable. Need inspiration? How about improved-octo-fiesta?

Description (optional): This is EXP 1

[✓ Public] Anyone on the internet can see this repository. You choose who can commit.

Private: You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

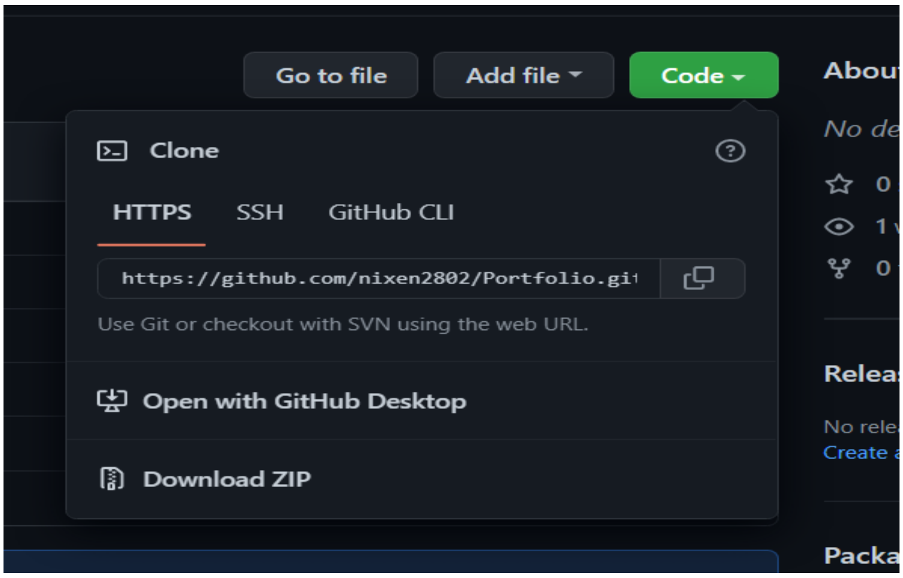
[✓] Add a README file
This is where you can write a long description for your project. Learn more.

Add .gitignore
Choose which files not to track from a list of templates. Learn more.

.gitignore template: None

5) Click Create repository. You'll be taken to your main repository page, which lists your files.

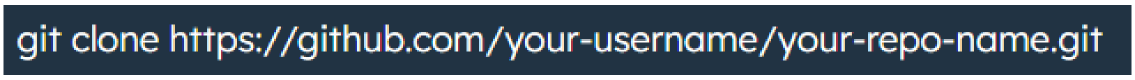
6) You'll now create a local copy of your GitHub repository (or in GitHub terms, "clone" your repository) where you'll edit your files and push your changes. On your main repository page, click the green Code button, then copy the HTTPS URL of your repository.



https://github.com/nixen2802/Portfolio.git

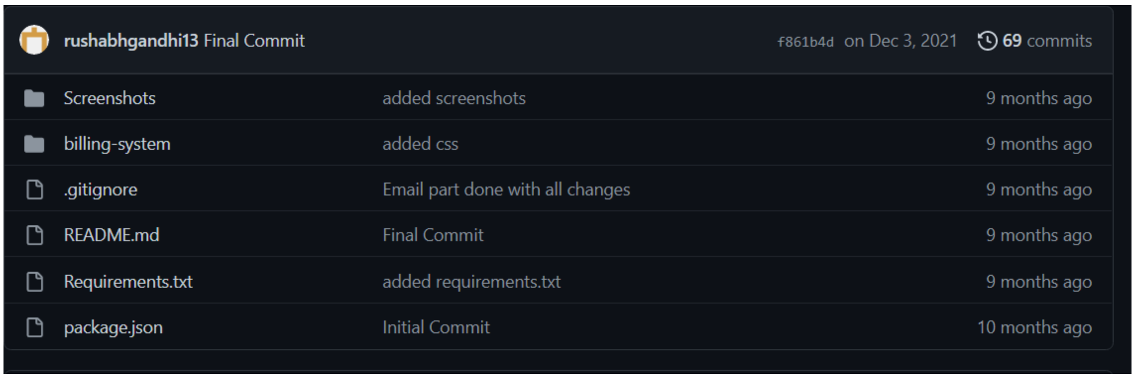
Use Git or checkout with SVN using the web URL.

7) Open your terminal and navigate to the directory you want to place your repository copy.
8) In the terminal, enter git clone. After this paste in the repository URL that you previously copied. Your command should look like this:



```
git clone https://github.com/your-username/your-repo-name.git
```

9) Press Enter to clone the repository. You'll see a new file added to your local filesystem with your repository's name. If you open this file, you'll see it contains the files in your GitHub repository. These are copied versions of your repository's files that you can edit and then push back to your repository.
10) Now you can add multiple files in the created folder.
11) After adding multiple files to add all the files to the git you should navigate to the folder in your terminal and use the command **git add .**(to add all the modified files on the github repository) to stage all the files onto the github.
12) In the terminal, type **git commit -m "modified something"** and press Enter. This commits your changes to the changelog. The text in quotes is a comment briefly describing the purpose of the commit.
13) Type **git push origin branch_name** in the terminal and press Enter.
14) Back in your GitHub repository, you'll see your new file added.

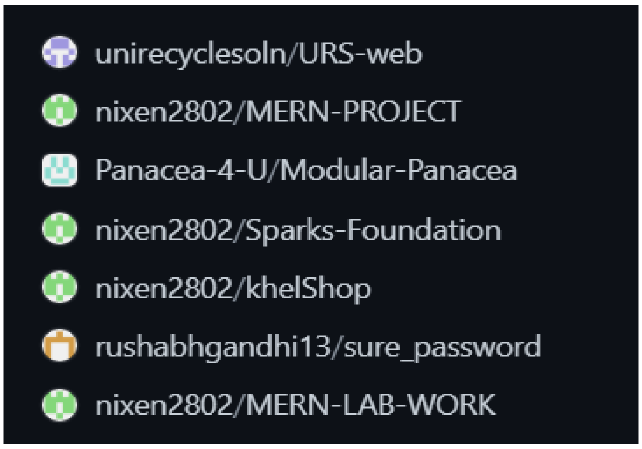


rushabhgandhi13 Final Commit f861b4d on Dec 3, 2021 69 commits

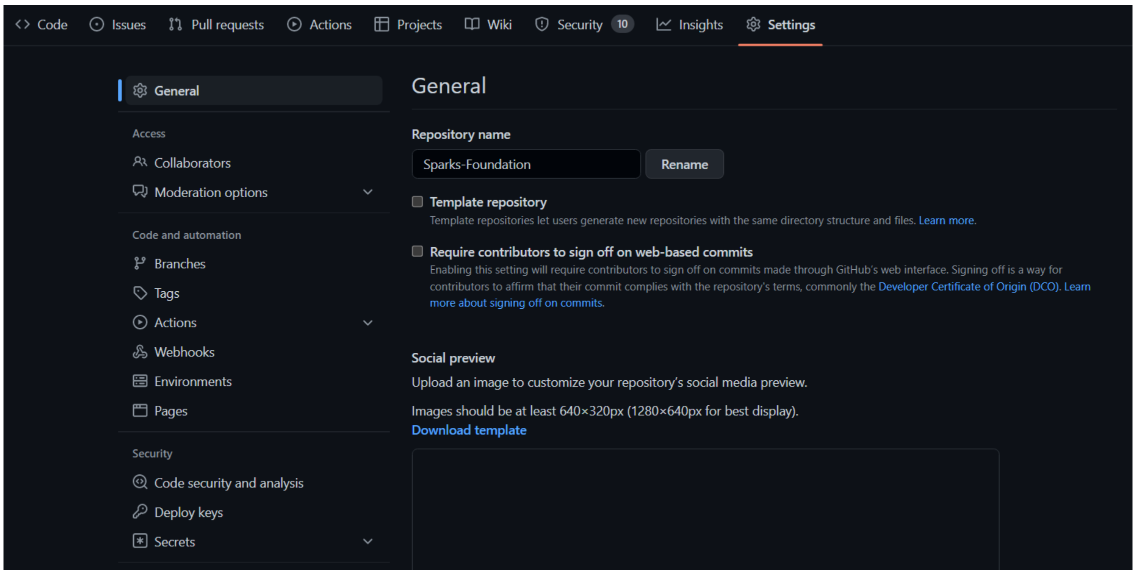
Screenshots	added screenshots	9 months ago
billing-system	added css	9 months ago
.gitignore	Email part done with all changes	9 months ago
README.md	Final Commit	9 months ago
Requirements.txt	added requirements.txt	9 months ago
package.json	Initial Commit	10 months ago

Deploying a website using github:

1) Go to your website's repository. After you've logged in, go to the repository on the left sidebar and select the one you want to publish.


unirecyclesoln/URS-web
nixen2802/MERN-PROJECT
Panacea-4-U/Modular-Panacea
nixen2802/Sparks-Foundation
nixen2802/khelShop
rushabhgandhi13/sure_password
nixen2802/MERN-LAB-WORK

2) Select the settings. In your repository, click the Settings link, and it will take you to the repository's settings page.



General

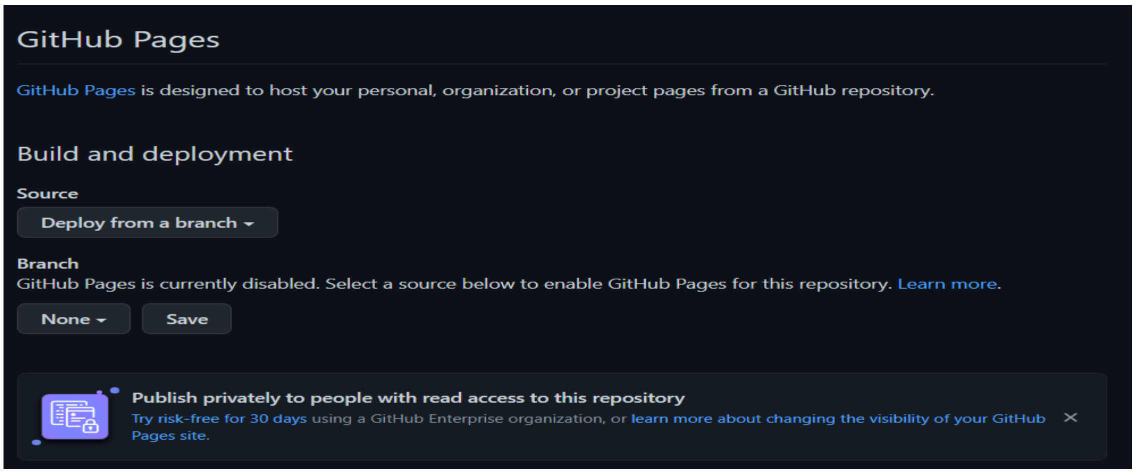
Repository name: Sparks-Foundation

Template repository
Template repositories let users generate new repositories with the same directory structure and files. Learn more.

Require contributors to sign off on web-based commits
Enabling this setting will require contributors to sign off on commits made through GitHub's web interface. Signing off is a way for contributors to affirm that their commit complies with the repository's terms, commonly the Developer Certificate of Origin (DCO). Learn more about signing off on commits.

Social preview
Upload an image to customize your repository's social media preview.
Images should be at least 640×320px (1280×640px for best display).
Download template

3) Go to GitHub Pages. When you're in a repository's settings, scroll down a bit until you see the Pages link on the left sidebar. Click it, and it will lead you to GitHub Pages.



GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

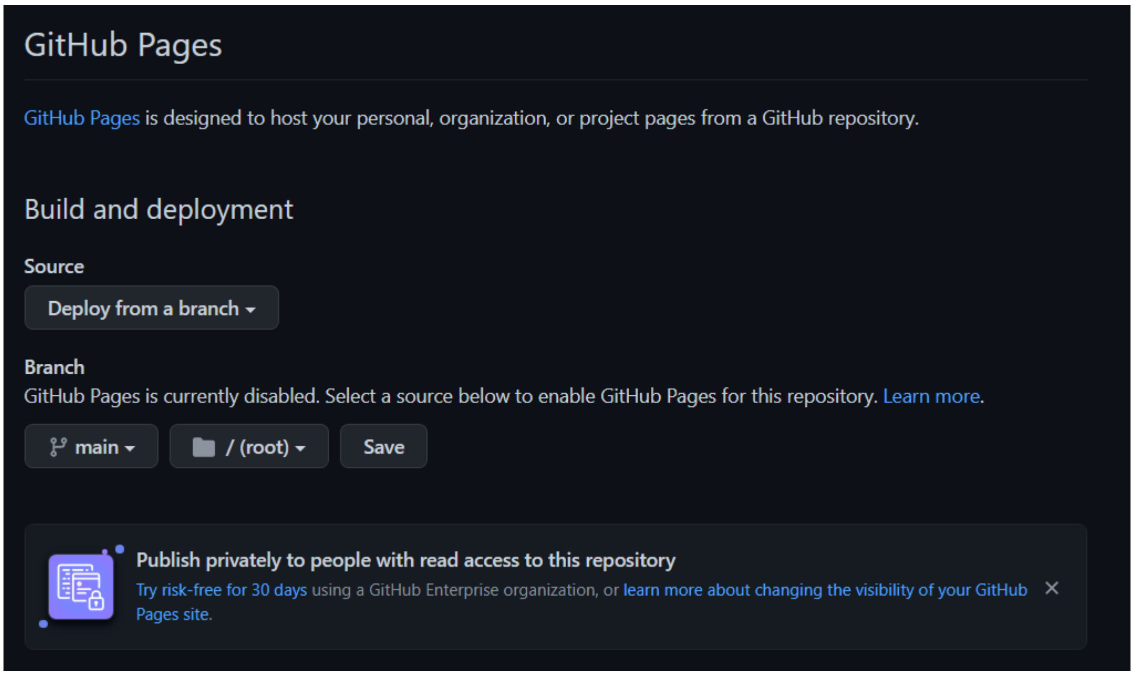
Build and deployment

Source: Deploy from a branch

Branch
GitHub Pages is currently disabled. Select a source below to enable GitHub Pages for this repository. Learn more.

Publish privately to people with read access to this repository
Try risk-free for 30 days using a GitHub Enterprise organization, or learn more about changing the visibility of your GitHub Pages site.

4) Select the branch. In the source section, click the dropdown and select the master branch and save it. Depending on how you name it, it can be master or main.



GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Build and deployment

Source

Deploy from a branch

Branch

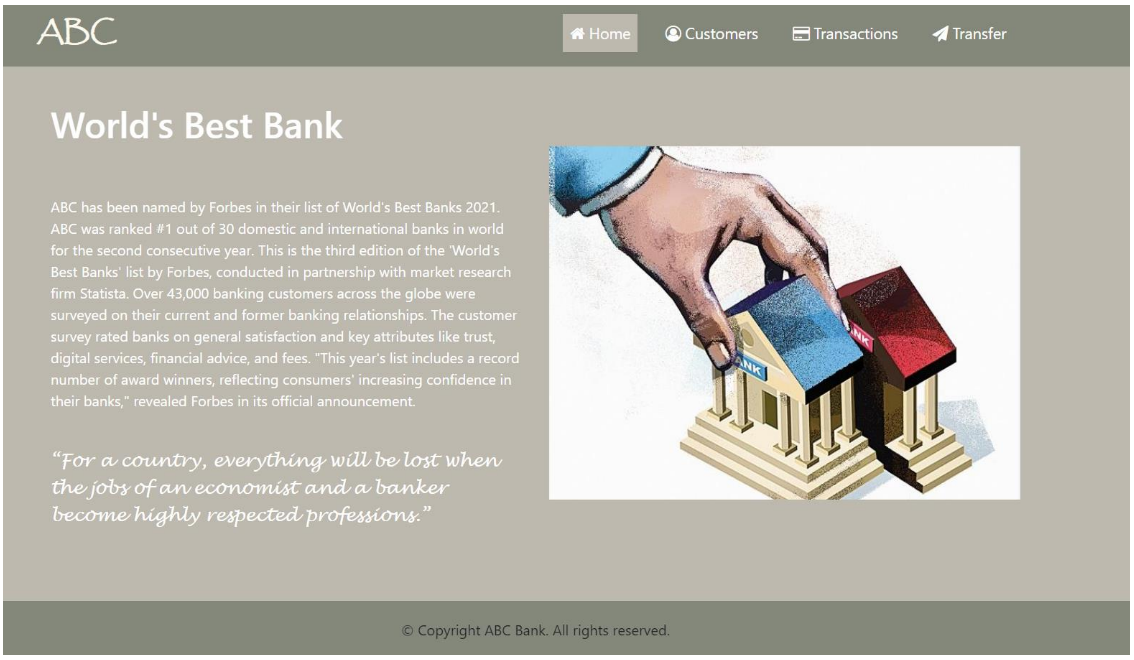
GitHub Pages is currently disabled. Select a source below to enable GitHub Pages for this repository. Learn more.

main / (root) Save

Publish privately to people with read access to this repository

Try risk-free for 30 days using a GitHub Enterprise organization, or learn more about changing the visibility of your GitHub Pages site.

5) Your website will be published, and it will take only a minute or so to complete the process. Refresh the page, and you will see a link to your newly published website.



ABC

Home Customers Transactions Transfer

World's Best Bank

ABC has been named by Forbes in their list of World's Best Banks 2021. ABC was ranked #1 out of 30 domestic and international banks in world for the second consecutive year. This is the third edition of the 'World's Best Banks' list by Forbes, conducted in partnership with market research firm Statista. Over 43,000 banking customers across the globe were surveyed on their current and former banking relationships. The customer survey rated banks on general satisfaction and key attributes like trust, digital services, financial advice, and fees. "This year's list includes a record number of award winners, reflecting consumers' increasing confidence in their banks," revealed Forbes in its official announcement.

"For a country, everything will be lost when the jobs of an economist and a banker become highly respected professions."

© Copyright ABC Bank. All rights reserved.

There are many other features are available in github they are as follows:

1) **Open a Pull Request:** In order for any branch to be merged into another person’s branch, you must open a pull request. A pull request is GitHub’s way of notifying relevant parties about your request to incorporate changes into their branch. A pull request will show in red and green the differences of the content between branches. You can make a pull request any time you complete a commit.
2) **Merge Your Pull Request:** Version control systems are all about managing contributions between multiple distributed authors (usually developers). Sometimes multiple developers may try to edit the same content. If Developer A tries to edit code that Developer B is editing a conflict may occur. To alleviate the occurrence of conflicts developers will work in separate isolated branches. The git merge command's primary responsibility is to combine separate branches and resolve any conflicting edits.
3) **Deployment of a project:** Users can use github to host their website, without paying anything at all, using GitHub Pages. The deployed website will be having the URL in the form of “your_custom_name.github.io”.
4) **Copilot:** GitHub Copilot uses AI to convert code comments into code. Based on the code comment input, Copilot provides suggestions on coding options. Then, developers select which suggestion to use or override the AI-generated suggestion. Copilot can also generate entire functions from one comment. This GitHub feature may be useful for things such as pair programming, supplementing development teams or performing code conversion projects. Teams should consider the feature to see if it improves the code, while reducing costs.

3. Explain the Importance of the approach followed by you

➔ Github has started moving from monolithic to microservices architectures both have their advantages and disadvantages:

1) In a monolithic environment, it’s easier to get up and running faster, without having to worry about complex dependencies and pulling in all the right pieces. A new Hubber can get GitHub up and running on their local machine within hours. There is some code-level simplicity in a monolith as well.

2) Additionally, because everyone is working in a shared tech stack and has familiarity with the same code base, it’s easier to move people and teams around to work on different features within the monolith and push towards a more global prioritization of features.

3) Microservices architecture makes smaller services also mean easier to read code, quicker ramp-up time, and easier troubleshooting within that code base.

4) Good architecture starts with modularity. The first step towards breaking up a monolith is to think about the separation of code and data based on feature functionalities. This can be done within the monolith before physically separating them in a microservices environment.

5) Getting data separation right is a cornerstone in migrating from a monolithic architecture to microservices.

6) After exploring the importance of data separation the focus was on how to lay the groundwork for extracting services out of the monolith.

7) Monitoring, CI/CD, and containerization are not new concepts, but making the necessary operational changes to support the transformation from monolith to microservices can yield significant time savings and help expedite the transition towards microservices.

8) Going from monolith to microservices is a major paradigm shift. Both the software development process and the actual code base will look significantly different going through this transition. There are two ways that services communicate with one another - synchronously and asynchronously. With synchronous communications, the client sends a request and waits for a response from the server. With asynchronous communications, the client sends a message without waiting for a response and each message can be processed by multiple receivers.

Conclusion:- Understood the importance of cloud computing and cloud computing architecture. Explored different types of cloud services such as SaaS, PaaS, and IaaS. Tested and used different features available on a SaaS product, Github and also understood the architecture of Github and how things internally work on Github.