Contents

[Introduction to ICT: 2](#_Toc155435030)

[Communication technologies: 3](#_Toc155435031)

[The Internet and World Wide Web: 4](#_Toc155435032)

[Collaboration Tools: 5](#_Toc155435033)

[**1.** Collaboration on interdisciplinary level (DevOps): 5](#_Toc155435034)

[**2.** Collaboration at the individual level: 6](#_Toc155435035)

[Operating Systems: 7](#_Toc155435036)

[Common Types of Operating Systems: 7](#_Toc155435037)

[Functionality and features overview of OS: 9](#_Toc155435038)

[Comparison of different OS capabilities: 10](#_Toc155435039)

[Programming languages: 11](#_Toc155435040)

[**1.** Programming Languages Classification: 11](#_Toc155435041)

[2. A Glimpse of Popular Languages: 11](#_Toc155435042)

[Version Control with Git and GitHub: 12](#_Toc155435043)

Introduction to ICT:  
 -ICT, which often stands for Information and communication technology is a a diverse set of technological tools and resources used to transmit, store, create, share and exchange information electronically.

ICT has evolved dramatically in the recent decades to become an important part of modern society. The prevalence of computers, mobile devices, the internet, and communication systems has led to an unlimited access to information and shaped people’s lives, work, and interactions. Some key unique aspects of ICT include:

* Innovation: The speed of ICT innovation continues to accelerate leading to ever more advanced capabilities in terms of connectivity, bandwidth, processing power, and applications. The continuous release of new technologies makes keeping up with and adopting them a never-ending aspect.
* Network effects: Many ICT systems exhibit powerful network effects where value grows exponentially as more users join. The global spread of the internet is a prime example of network effects in action.
* New Modes of Communication: ICT has enabled new forms of human communication and collaboration, beating the time and geography barriers with the introduction of E-mails, messaging, voice and video cheats and Social Medias (e.g., Facebook, Instagram, Twitter etc.)
* The impact on Business, Work and Society: From how business is conducted to how people socialize and govern ICT has had a transformative and disruptive impact on modern human activity and organization. Overall, the full implications of ICT are still unfolding.

-ICT is very important today. It helps businesses work faster and better. It has changed how people work as it’s easier now for people to do freelance and independent work. They don’t need a traditional job as they can do most of their work at home and communicate with and do work for clients from anywhere. Companies can use ICT to work together from different places. ICT lets businesses sell and advertise online. For schools, ICT provides new ways to teach and learn. Students can use technology like computers and tablets in classrooms. They can also take online classes and find information on the internet. ICT helps people communicate quickly with messaging and video chat. Friends and families can connect even if they live far apart. ICT makes many things possible that were thought impossible to achieve before. It has changed how we live, work, learn and communicate. New ICT keeps getting developed. Using it more will help society keep improving.

-ICT started with basics things like the telephone (1876) and radio (1895). Later came computers (1940s) and the internet (1960s). In the beginning, only simple websites and emails were possible. After technology kept advancing, people could use faster internet on smartphones and laptops. This allowed big changes like social media, online shopping and video chat. Today, ICT includes very complex things like artificial intelligence and virtual reality. All these new ICT innovations happened very fast over the last 150 years. It has come along way and is still currently evolving and going to go through multiple innovations in the future.

# Communication technologies:

-ICT has enabled innovative communication tools and platforms. Prominent communication technologies shaping modern digital interaction include:

* **E-Mail:** Emails revolutionized digital communication by allowing messages to be sent electronically between networked computers. Users have personalized email addresses that route messages to their inboxes. Email clients and webmail services manage composing, sending, and organizing emails. Communication via email is asynchronous allowing flexibility in when messages are exchanged. Email protocols like SMTP handle routing envelopes while IMAP and POP download messages to inboxes.
* **Instant Messaging:** Instant messaging (IM) enables real-time communication between users. Popular IM platforms include WhatsApp Facebook, Messenger, and WeChat. Presence indicators show who is online/available for chatting. IM allows text conversations alone or integrated with file, image, and video sharing, group conversations are possible in IM. The immediacy of IM differentiates it from asynchronous email.
* **Voice over IP:** Voice over IP (VoIP) transmits voice communications over the internet rather than traditional telephone networks. Skype popularized VoIP services for consumers. Business VoIP services also exist. VoIP can be integrated with instant messaging clients. Call quality can vary based on internet connectivity. However, VoIP offers flexibility and cost saving versus analog telephony.
* **Video Conferencing:** Video conferencing uses dedicated platforms like Zoom, Microsoft teams and Webex for real-time video calls. Features include screen sharing call recording, virtual backgrounds, and support for meeting of various sizes. Video conferencing has enabled new norms of remote work and learning during the COVID pandemic. Easy to use software and affordable computer webcams have made quality video conferencing accessible to the world.
* **Social Media:** social media allows users to connect by sharing contents like text, photos, videos. Leading platforms are Facebook, Instagram, Twitter (also known as X now), TikTok and Snapchat. Social media enabled forming communities online to share interests or keep in touch. Key features include newsfeeds, messaging, profiles, notifications. Adoption of social media has grown enormously to billions of users worldwide.

# The Internet and World Wide Web:

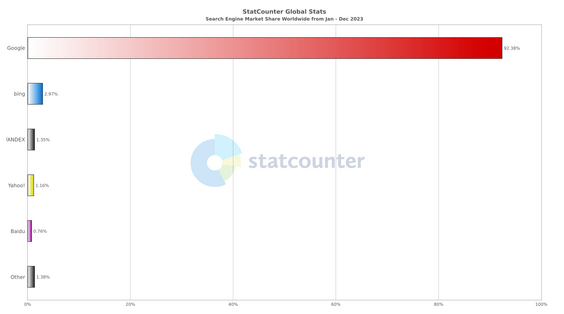
**-**In the modern world, the World Wide Web and the internet, in general, have opened up unprecedented opportunities for global interaction and knowledge sharing. Over time, however, the internet has become a cluttered space with a plethora of unwanted content that may distract us from our intended searches or interests. Consequently, search engines were developed, equipped with powerful algorithms and computing capabilities.

**-**A search engine is a software that operates in the browser, providing a readable (text/plain) communication with the World Wide Web, distancing users from the unreadable protocol requests. This brings us to the question of why search engines were created. The first web search engine, named Archie, was developed in 1990 by Alan Emtage, a student at McGill University in Montreal, Canada. It was primarily implemented for indexing FTP archived files. In contrast, modern search engines like Google, Bing, and Yahoo focus on web-based content indexing, employing complex algorithms such as PageRank and TF-IDF to evaluate the importance of web pages based on their incoming links.

**-**The web utilizes protocols like HTTP/HTTPS (Hypertext Transfer Protocol) to communicate and establish server/client connections. HTTP typically uses port 80 for communication, employing text/plain to transfer information and cookies. On the other hand, HTTPS uses port 443 and encrypts data with SSL/TLS encryption to prevent network sniffing. Thus, not only is the ability to transfer data crucial, but also the security of that data, especially with the advent of user interaction and input on websites

-Initially, the web comprised static HTML pages that users could read akin to traditional newspapers. However, with the introduction of Web 2.0, which brought dynamic and maintained connections with AJAX (Asynchronous JavaScript and XML), web pages could update content asynchronously by exchanging small amounts of data behind the scenes. This prevented delays caused by slow internet connections at the time. Now, with Web 3.0, privacy has been enhanced, and new terms such as blockchain, linked data, and interconnected platforms have emerged.

-The primary role of search engines is to filter unwanted data. Websites typically use a robots.txt file to indicate what content should not appear on Search Engine Results Pages (SERPs) when a user searches for that website. Today, search engines are equipped with AI tools like Google BERT and Bing Chat. All these features influence the popularity of a search engine and its market presence. As of 2023, Google dominates the search engine market with a staggering 92.38% market share, according to statcounter.com.



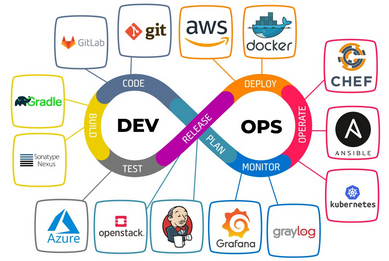
-From the server and company perspective, various technologies are employed to ensure faster and more reliable server responses. These technologies contribute to optimizing performance and enhancing user experience. Some notable examples include:

* Content Delivery Networks (CDN’s) like Cloudflare, Akamai, Amazon, CloudFront and many more.
* Keeping protocols that we talked about earlier (HTTP/2 and HTTP/3)
* Caching mechanisms involve storing frequently displayed data on the client device such as images, charts, footers and headers to optimize performance.
* WebSocket: Websockets enable real-time communication between server and the client, facilitating interactive features in Web 2.0 applications.
* Distributed Databases: Web servers often rely on distributed databases for improved data delivery, availability, fault and tolerance and scalability.
* Edge Computing: Placing computing resources closer to the end-user at the network edge reduces latency and enhance overall performance of web applications.

# Collaboration Tools:

1. Collaboration on interdisciplinary level (DevOps):  
    -DevOps, a portmanteau of development and operations, represents a transformative approach to software development and its operations. In the realm of Information and Communication Technology (ICT), the adoption of DevOps methodologies has significantly impacted industrial processes, fostering collaboration, agility, and efficiency. This approach is founded on several key principles aimed at bridging the gap between development and operations teams, fostering a culture of collaboration and continuous improvement. These principles encompass automation, continuous integration, continuous delivery/deployment (CI/CD), and monitoring.
   * **Continuous Integration (CI):** CI involves regularly integrating code changes into a shared repository, allowing for the early detection of integration issues. This practice enables teams to work cohesively and detect/address potential conflicts swiftly, ensuring a more stable and maintainable codebase.
   * **Continuous Delivery/Delivery (CD**): CD extends the benefits of CI by automating the release and deployment processes. This results in a more streamlined and predictable delivery pipeline, facilitating the rapid and reliable release of software updates for the ICT sector. This means faster adaptation to market demands and the ability to provide more responsive and up-to-date solutions.

-DevOps promotes a culture of collaboration and open communication between development and operations teams. This cultural shift encourages the sharing of knowledge, experiences, and responsibilities, breaking down silos and fostering a more holistic understanding of the entire software delivery lifecycle.



1. Collaboration at the individual level:  
    -The evolution of IT has given rise to infrastructure projects that require simultaneous collaboration from multiple individuals. To emulate the professionalism of companies, individuals must engage in collaborative efforts within the realm of IT. This involves adopting effective collaboration tools, fostering open communication, and aligning workflows to mirror the organized and efficient practices seen in professional corporate environments. We can observe these principles in many open-source solutions challenging products from established corporations. An exemplary product/tool within this context is **Git**.   
   Git is a distributed version control system (VCS) that allows multiple people to work on projects simultaneously. It efficiently tracks changes in source code during software development, enabling collaboration and version management. It was created by Linus Torvalds, who is known for his contributions to well-known projects like the Linux kernel, an open-source Unix-like operating system kernel, in 1991 while he was a student at the University of Helsinki. For the same reasons discussed earlier, 'to create open-source solutions to challenge corporate products,' in this case, Windows from Microsoft and the classic Mac OS from Apple. Git was primarily published to handle the demands of a huge project like the Linux kernel. The development of Git marked a departure from centralized version control systems. Its decentralized nature allowed developers to work offline, make local commits, and collaborate seamlessly. Over the years, Git gained widespread adoption in the software development community and became the go-to version control system for many projects, both large and small.  
   Git has been implemented in many GUI (graphical user interface) solutions such as:

* GitHub: GitHub is a web-based platform built around Git. It was established in 2008 by Tom Preston-Werner, Chris Wanstrath, and PJ Hyett and is owned by Microsoft for version control and collaboration. GitHub provides a graphical interface for Git repositories and extends Git's functionality. It adds features like pull requests, issues, and project management tools to enhance collaboration among developers. Additionally, it includes social features, making it a hub for open-source projects and facilitating community involvement. GitHub also offers some additional paid features
* GitLab: GitLab is another web-based platform that provides Git repository management and CI/CD (Continuous Integration/Continuous Deployment) capabilities. It also offers features similar to GitHub, such as merge requests (equivalent to pull requests in Git), issue tracking, and a wiki. While GitHub is owned by Microsoft, GitLab is open-source.

These two are by far the most popular implementations of Git today but there are many more like Bitbucket, Azure, GitKraken etc.

# Operating Systems:

- An operating system (OS) is a fundamental software component that serves as an intermediary between computer hardware and user applications. It provides a cohesive and efficient environment for managing resources, executing programs, and facilitating communication between hardware and software components.  
 -Operating systems are built over a kernel. A kernel is the only way that humans can communicate directly with the hardware they use, with syscalls. Operating systems and their interfaces are created to facilitate these communications.

## Common Types of Operating Systems:

* + **Windows:** An operating system developed by Microsoft and stands as a cornerstone in the landscape of personal computing. Here’s an overview of key aspects of the Windows operating system:
    - **Graphical User Interface (GUI):** Windows is renowned for its user-friendly graphical interface featuring the Start menu, taskbar and desktop icons.
    - **Software Compatibility:** Widely known for extensive software compatibility, supporting a vast array of applications and games.
    - **Security Measures:** Incorporates security features like Windows Defender (Anti-Virus), BitLocker (Disk Encryption) and User Account Control (UAC) for system integrity.
    - **File System:** Utilizes NTFS (New Technology File System) for efficient organization and storage.
    - **Microsoft Store:** Offers a centralized platform for users to download and install applications, including both traditional software and Universal Windows Platform (UWP) applications.
    - **DirectX and Gaming:** Supports DirectX, making it a popular platform for gaming with a wide range of compatible games and applications.
  + **MacOS:** Previously known as Mac OS X and OS X, macOS is the operating system developed by Apple Inc. exclusively for Macintosh computers. The first version, Mac OS X, was introduced in 2001, with subsequent updates and enhancements leading to the current macOS versions. Here's an overview of key aspects of the macOS operating system:
    - **User Interface and Design:** macOS is renowned for its sleek and intuitive user interface featuring a top menu bar dock for app shortcuts and the Finder for file management. The design emphasizes simplicity, elegance and ease of use.
    - **Unix-Based Foundation:** macOS is built on a Unix-based foundation, providing a stable and secure environment for users. This foundation allows macOS to offer a robust command-line interface and compatibility with Unix software.
    - **App Store and Ecosystem:** The Mac App Store is the official platform for downloading and installing applications on macOS. macOS is tightly integrated into the Apple ecosystem connecting with iCloud for seamless data synchronization across devices.
    - **Security Features:** macOS incorporates various security features, including Gatekeeper for app verification FileVault for disk encryption and XProtect for malware detection.
    - **Creative and Professional Applications:** macOS is widely adopted by creative professionals due to its compatibility with industry-standard applications like Final Cut Pro, Logic Pro and Adobe Creative Cloud.
  + **Linux:** A powerful and versatile open-source operating system kernel that serves as the foundation for a variety of operating systems. Initially developed by Linus Torvalds in 1991. Here’s an overview of key aspects of the Linux operating system:
    - **Open Source:** Linux is distributed under the GNU General Public License (GPL) making its source code freely available. This openness encourages collaboration and continuous improvement.
    - **Security Features:** Linux is inherently secure, with features such as user privilege management, file permission systems and robust access controls. Regular updates help address security vulnerabilities promptly.
    - **Command-Line Interface (CLI):** Linux provides a powerful command-line interface (CLI) allowing users to execute tasks efficiently and customize system configurations.
    - **Community Collaborations:** The Linux community plays a crucial role in its development and improvement. Users, developers and organizations contribute to its ongoing evolution.
    - **Kernel-Based:** The Linux kernel is at the core of the operating system managing system resources and facilitating communication between hardware and software components.
  + **iOS:** A mobile operating system developed by Apple Inc. for its line of iPhones, iPads and iPod Touch devices. It is a closed-source operating system designed to work exclusively with Apple’s hardware. Here’s an overview of key aspects of the iOS operating system:
    - **App Store:** The App Store is the official marketplace for iOS applications. Users can download and install a wide variety of apps, ranging from games and productivity tools to social media and entertainment.
    - **Security:** iOS is known for its robust security measures. This includes the App Store’s stringent app review process, sandboxing of apps and hardware-level security features such as Face ID and Touch ID for user authentication.
    - **Updates:** Apple regularly releases iOS updates to introduce new features, enhance security and fix bugs. Users can easily update their devices over the air.
    - **iCloud Integration:** iCloud is Apple’s cloud service that seamlessly integrates with iOS. It provides features such as automatic device backup, synchronized data across devices and storage for photos and documents.
    - **Siri:** Siri is Apple’s virtual assistant integrated into iOS, offering voice-activated assistance for tasks, inquiries and device control.
    - **Multitasking:** iOS supports multitasking, allowing users to run multiple apps simultaneously. This includes features like split-screen mode on iPads.
    - **Family Sharing:** Family sharing allows users to share purchased apps, music and other content with family members. It also includes parental controls for managing children’s devices.
    - **Health and Fitness:** iOS includes the Health app, which gathers and organizes health and fitness data from various sources, promoting a comprehensive approach to wellness.
  + **Android:** A mobile operating system developed by the Open Handset Alliance, a consortium of developers led by Google. It is designed for a wide range of touchscreen devices such as smartphones, tablets, smartwatches, TVs and many more. Here’s an overview of key aspects of the Android operating system:
    - **Open-Source Architecture:** Android is built on a modified Linux kernel, providing an open-source platform that encourages innovation and collaboration.
    - **Application Ecosystem:** The Google Play Store hosts a vast array of applications, offering users a diverse selection of software to meet their needs.
    - **Regular Updates and Iterations:** Android undergoes frequent updates, introducing new features, performance improvements and security enhancements to keep the operating system current.
    - **Developer-Friendly Environment:** The Android SDK enables developers to create applications using popular programming languages like Java, Kotlin and C++, fostering a vibrant app development community.

## Functionality and features overview of OS:

-Operating systems serve as the backbone of computing devices, providing essential functionalities and features to ensure a seamless user experience. Here’s a summary of the key aspects:

* + **User Interface (UI):** Operating systems offer graphical interfaces with elements like icons and menus, accommodating various input methods for user interaction.
  + **File Management:** Efficient organization and handling of files through tools like File Explorer or Finder, supporting operations like copying, pasting, moving and searching.
  + **Security:** Protects the system through user authentication, firewalls, antivirus tools and encryption to safeguard against unauthorized access and malware.
  + **Networking:** Supports connectivity with local and internet networks, providing options for configuring Wi-Fi, Ethernet and other network settings.
  + **System settings:** Provides a centralized location for configuring preferences including user accounts, permissions and other system settings.

## Comparison of different OS capabilities:

-This involves comparing the capabilities of different operating systems by considering various factors such as user interface, customization options, security, app ecosystem and more. Let’s compare three major operating systems: Android, iOS and Windows.

* + **User Interface**
    - **Android:** Highly customizable interface, allowing users to personalize home screens with widgets and themes.
    - **iOS:** Offers a clean and consistent interface across all Apple devices, prioritizing a uniform user experience.
    - **Windows:** Provides a customizable Start menu and Live Tiles for quick access to information with varying consistency between desktop and mobile version.
  + **App Ecosystem**
    - **Android:** Google Play Store hosts a vast array of applications and Android often receives new apps first due to its open nature.
    - **iOS:** The curated App Store ensure generally higher app quality and some apps may have exclusive releases on iOS.
    - **Windows:** The Microsoft Store offers applications with a focus on universal apps that work across different Windows platforms.
  + **Customization**
    - **Android:** Offers a high level of customization through third-party launchers, widgets and system tweaks allowing users to choose default apps.
    - **iOS:** Limited customization compared to Android with a more controlled user experience, some customization options are available for the home screen default apps.
    - **Windows:** Provides customization options for the Start menu and Live Tiles with more extensive options on the desktop version compared to the mobile one.
  + **Security**
    - **Android:** Security concerns due to its open nature but Google implements various security features and regular security updates are provided.
    - **iOS:** Known for a strong security architecture, prompt updates and a tightly controlled App store, reducing the risks of malware.
    - **Windows:** Generally secure, with less vulnerability to malware compared to Android and iOS and regular security updates for Windows operating systems.
  + **Device Diversity**
    - **Android:** Widely available on devices from various manufacturers, offering diversity in hardware specifications and user interfaces.
    - **iOS:** Exclusive to Apple devices (iPhone, iPad, iPod Touch etc.) providing a consistent experience but limiting device diversity.
    - **Windows:** Used on various devices, including PCs, tablets and smartphones. Device diversity depends on the specific Windows version.

-In summary, the choice between Android, iOS and Windows or any other OS depends on the individual preferences, device requirements and ecosystem preferences. Android offers high customization and device diversity; iOS provides a consistent and secure experience across Apple devices while Windows is known for its integration with PCs and a customizable interface.

# Programming languages:

-Embarking on an exploration of programming languages, we'll delve into the diverse landscape of widely-used languages, each possessing distinct traits and making significant contributions to the tech realm. The first-ever programming language, apart from machine code (composed of 0’s and 1’s), is assembly language. It communicates directly with the hardware through the kernel and system calls (SYSCALLS), consisting of instructions that are not easily readable to the average person or even an average tech enthusiast. Consequently, many higher-level languages were created to simplify development. Examples include Python, known for its user-friendly syntax, JavaScript's ubiquitous role in web development and browser interaction, C#'s integration prowess, and Java's cross-domain versatility. Each language weaves its narrative of innovation and practical application.

1. Programming Languages Classification: -Programming languages can be classified into several categories based on their characteristics and usage. One primary classification is between high-level and low-level languages. High-level languages are more abstract and user-friendly, often providing better readability and ease of development. Low-level languages, like Assembly or machine code, are closer to the hardware and offer more direct control over the computer's resources. The higher the language level, the more readable it is for developers and the easier it is to code with, facilitating communication between the developer and the application or project and hardware resources. On the other hand, low-level languages provide the ability to use these resources directly, such as addressing memory with registers and taking control of data types.
2. A Glimpse of Popular Languages:

* **Python:** It is a high-level programming language renowned for its straightforward syntax and readability. Python stands out as an all-purpose language finding applications in web development, machine learning and automation. It’s extensive libraries and vibrant community make it a favorite among developers seeking efficiency and expressiveness.
* **JavaScript:** Serving as the backbone of web development, JavaScript empowers dynamic, client-side interactions. Universally supported by browser, it plays a pivotal role in creating responsive user interfaces. With the rise of Node.js, JavaScript extends its influence to server-side development.
* **C#:** Crafted by Microsoft, C# is a formidable, statically-typed language used in developing Windows applications games development using Unity and enterprise-level software. Its seamless integration with the .NET framework enhances interoperability and boosts productivity.
* **Java:** Prioritizing portability and robustness Java is a versatile language employed in enterprise solutions Android app development and large-scale systems. Its “Write once, run anywhere” philosophy makes it well-suited for cross-platform development.

1. Navigating Web Development Frameworks:
   * **React:** Born out of Facebook’s ingenuity, React simplifies UI development through its component-based architecture. It excels in crafting interactive single-page applications by efficiently managing the state of user interfaces.
   * **Angular:** Hailing from Google, Angular is a comprehensive framework emphasizing modularity and maintainability. It provides a rich set of tools for constructing dynamic web applications, featuring two-way data binding and dependency injection.
   * **Django:** A high-level Python web framework, Django fosters swift development with a clean and pragmatic design. Following the Model-View-Controller (MVC) architectural pattern and incorporating on Object-Relational Mapping (ORM) system, it simplifies database interactions.
   * **Node.js:** Enabling server-side JavaScript execution, Node.js is crucial for scalable and real-time applications. Its event-driven, non-blocking I/O model facilitates the creation of fast and lightweight network applications transforming JavaScript into a full-stack language.
2. Unpacking Frameworks and Their Role in Programming: -Frameworks server as guiding structures for developers, offering a structured approach to application building. They encapsulate best practices, provide reusable components and streamline routine tasks. By abstracting complex functionalities, frameworks empower programmers to tackle specific challenges without reinventing the wheel. This not only accelerates development but also ensures code consistency and scalability resulting in more maintainable and efficient software solutions.

# Version Control with Git and GitHub:

-The integration of Git and GitHub in version control plays a crucial role in enhancing collaborative efforts, ensuring precise code management and optimizing the efficiency of software development life cycles.

1. Benefits of Version Control Systems like Git:Git, being a distributed version control system, offers meticulous tracking of code evolution facilitates collaborative efforts among team members and establishes code stability through systematic versioning.
2. GitHub Overview for Source Code Hosting and Collaboration**:** GitHub, a cornerstone platform, serves as a repository for organized code management. It supports issue tracking for effective bug governance and enables pull requests for detailed code review and collaborative refinement.
3. Using Git Locally with:
   * **GitHub For Teams:** The amalgamation of Git’s localized features with GitHub streamlines the process of code modification. Developers can efficiently clone repositories create branches for independent coding tasks and push updates to GitHub for collaborative refinement.
   * **Gitea:** It is a self-hosted Git service, originating from a 2016 fork of Gogs. It offers lightweight and open-source version control comparable to platforms like GitHub or GitLab. Known for simplicity and user-friendliness, Gitea supports repository management, issue tracking and collaboration tools making it suitable for individual developers and small teams. Its emphasis on performance and minimal resource requirements makes it a versatile choice for self-hosted Git solutions, catering to collaborative coding projects and efficient version control.
4. Workflows like Branching Strategies, Pull Requests:Git’s branching strategies provide teams with systematic approaches for managing features development, bug fixes and releases. GitHub’s pull requests allow developers to propose code changes, initiate comprehensive code reviews and seamlessly integrate new features into the existing codebase. These workflows contribute to a structured development process, ensuring collaboration and maintaining code quality.

