ChatGPT

Comprehensive Map of Technology and the Internet

Introduction

Modern technology encompasses a vast ecosystem of hardware, software, networks and services that enable almost every aspect of daily life. The internet is built upon decades of innovation in computing, communications, data management and human-computer interaction. Because the field is broad, this guide provides a **high-level map** of major domains within technology (computer architecture, programming, networks, databases, cybersecurity, cloud computing, data science, artificial intelligence, blockchain, the internet of things, virtualization, human-computer interaction and immersive technologies). Each section offers:

- **Definition & significance** with citations from authoritative sources.
- **Subtopics** outlining deeper avenues for exploration.
- **Learning path** with approximate time commitments based on the standard definition of a credit hour: *a semester hour equals about one hour of classroom instruction and two hours of out-of-class work each week for a 15-week semester* [1]. A three-credit course therefore requires roughly **135 total hours of study**.
- **Recommended learning resources** (suggested open courses and textbooks).

Use this overview to survey the landscape and then dive deeper into the subjects that interest you. Times are estimates for a rigorous, university-level understanding but can be adjusted based on prior experience and pace.

High-level Overview of Key Domains

Domain	Concise description	Approx. learning time*
Computer Architecture & Hardware	Focuses on how a computer's components (CPU, memory, storage and I/O devices) communicate through electronic signals to perform input, processing and output [2].	~14–16 weeks (3 credits)
Programming & Software Development	Encompasses programming fundamentals, data structures and algorithms, software engineering practices and version control.	1–2 years for a broad foundation (several courses)
Operating Systems	System software that acts as an interface between hardware and users, manages resources and runs other programs [3].	~14–16 weeks (3 credits)
Networks & Internet	Networks connect multiple devices to share information [4]; the internet layers protocols like TCP/IP and HTTP. Includes web development.	~14–16 weeks for networking; another 14–16 weeks for web development

Domain	Concise description	Approx. learning time*
Databases & Data Management	A database management system is software for creating, protecting, reading, updating and deleting data in organized structures [5].	~14–16 weeks
Cybersecurity	The practice of protecting systems, networks and data from digital threats; aims to preserve confidentiality, integrity and availability [6].	~14–16 weeks
Cloud Computing & Virtualization	Cloud computing provides on-demand network access to shared configurable resources [7]; virtualization creates a virtual version of physical technology such as an OS, server or network [8].	~14–16 weeks
Data Science & Statistics	A field using scientific methods to extract knowledge and insights from data [9]; involves statistics, computing and domain expertise.	1–2 semesters
Machine Learning & AI	Machine learning is a subfield of artificial intelligence that gives computers the ability to learn without being explicitly programmed [10].	1–2 semesters
Blockchain & Decentralized Systems	A blockchain is a collaborative, tamper-resistant ledger where transaction records are grouped into blocks and linked via unique identifiers [11].	~8–12 weeks
Internet of Things (IoT)	The internet of things is a network of interrelated devices embedded with sensors and software that exchange data with each other and the cloud [12].	~8–12 weeks
Human-Computer Interaction (HCI) & UX	User Experience (UX) refers to the feeling users experience when using a product or system and the holistic journey they take [13]. HCI studies these interactions to design usable systems.	~8–12 weeks
Immersive Technologies (AR/VR)	Augmented Reality overlays digital information onto the real world, while Virtual Reality creates a completely artificial environment [14].	~6–8 weeks
Ethics, Privacy & Law	Addresses ethical considerations, data privacy, intellectual property and regulations affecting technology.	~6–8 weeks

*Based on standard credit-hour definitions [1]. Total time varies with depth of study.

Detailed Domain Descriptions and Learning Paths

1. Computer Architecture & Hardware

Definition & significance. Computer architecture defines **how components communicate through electronic signals to perform input, processing and output operations** [2]. It covers the design and organization of the CPU, memory, storage and I/O devices, and it influences system speed, functionality and reliability.

Key subtopics.

- **Digital logic & number systems.** Binary representation, Boolean algebra and combinational/ sequential circuits.
- **Microarchitecture.** CPU organization, registers, arithmetic logic units (ALUs), instruction pipelining, RISC vs. CISC architectures.
- **Memory hierarchy.** Caches, main memory, virtual memory and storage systems.
- **Instruction Set Architecture (ISA).** Assembly language, addressing modes and instruction formats.
- **Input/Output systems.** Interrupts, Direct Memory Access (DMA), buses and controllers.
- **Parallel architectures.** SIMD, MIMD and vector processors; multicore systems.

Learning path & time.

A typical **Computer Architecture** course is 3 credits (~135 hours) and spans about **14–16 weeks**. Recommended resources include:

- **Textbook:** *Computer Organization and Design* by Patterson & Hennessy.
- **OpenCourseWare:** MIT's *Computer Architecture* and *Digital Systems* courses.
- **Practical exercises:** Build simple circuits with logic simulators; implement assembly programs; analyze CPU performance.

2. Programming & Software Development

Programming is the foundation of technology. It involves learning languages, problem-solving, data structures, algorithms and best practices for building software.

2.1 Programming Fundamentals

Topics. Variables and data types, control structures (loops, conditionals), functions, recursion, basic object-oriented programming (classes, objects, inheritance), memory management and debugging.

Learning path. Start with an introductory programming course (Python or Java). A 3-credit course (~135 hours) covers these basics. Online options include **Harvard's CS50** and **MIT's Introduction to Computer Science and Programming**.

2.2 Data Structures & Algorithms

Definition. A **data structure is a way of organizing and storing data** so that it can be accessed and used efficiently [15]. Algorithms are step-by-step procedures for performing tasks. Mastery of data structures and algorithms (DSA) is crucial for efficient software.

Key subtopics.

- **Linear structures:** arrays, linked lists, stacks and queues.
- **Non-linear structures:** trees (binary trees, BSTs, heaps, B-trees), graphs (representation, traversal algorithms like BFS and DFS).
- **Hashing:** hash tables, collision resolution.
- **Algorithms:** sorting (quick sort, merge sort), searching, dynamic programming, greedy algorithms, recursion and divide-and-conquer strategies.
- **Complexity analysis:** Big-O notation to evaluate time and space requirements.

Learning path. A standard **Data Structures & Algorithms** course is often 4 credits (~180 hours) and spans a semester. Recommended resources:

- **Books:** *Introduction to Algorithms* by Cormen et al., *Data Structures and Algorithm Analysis* by Weiss.
- **Online:** Stanford's *Algorithms* Specialization (Coursera), *GeeksforGeeks* tutorials.
- **Practice:** Solve problems on coding platforms (LeetCode, HackerRank) to solidify understanding.

2.3 Software Engineering & Project Management

This area covers methods for designing, building, testing and maintaining software systems.

Topics. Software development life cycle (SDLC), requirements analysis, design patterns, testing (unit, integration, system), version control (e.g., Git), agile methodologies, documentation and code review.

Learning path. Two courses (~2 × 3 credits). Use textbooks like *Software Engineering* by Ian Sommerville and resources from the **IEEE Software Engineering Body of Knowledge (SWEBOK)**. Hands-on projects are essential—build and maintain medium-sized applications in teams.

2.4 DevOps & Continuous Delivery

Definition. DevOps is a set of **practices, tools and a cultural philosophy** that automate and integrate processes between software development and IT teams [16]. It emphasizes team empowerment, cross-team communication and collaboration, and automation.

Subtopics.

- **Version control & collaboration** (Git, branching strategies).
- **Continuous Integration/Continuous Delivery (CI/CD)** pipelines for automated testing and deployment.
- **Infrastructure as Code (IaC)** using tools like Terraform or Ansible.
- **Containerization** (Docker) and orchestration (Kubernetes).
- **Monitoring & observability:** logging, metrics, tracing.
- **DevSecOps:** integrating security into the development pipeline [17].

Learning path. One semester (~3 credits). Start by automating builds and tests; then deploy simple applications using CI/CD pipelines on cloud platforms. Resources: **Atlassian DevOps guides**, **Google's SRE book**, **Kelsey Hightower's Kubernetes tutorials**.

3. Operating Systems

Definition. An **operating system (OS)** acts as an interface between hardware and users, running at all times and providing an environment for programs to execute [18]. It assigns resources (memory, CPU, I/O) to processes, ensuring fairness and security [19].

Key subtopics.

- **Process management:** processes, threads, scheduling algorithms, context switching.
- **Memory management:** segmentation, paging, virtual memory and page replacement algorithms.
- **File systems:** directories, file allocation strategies, permissions.
- **Concurrency & synchronization:** race conditions, mutual exclusion, semaphores and monitors.
- **Device management:** I/O systems, buffering, interrupt handling.
- **Virtualization fundamentals:** OS-based virtualization and containerization.

Learning path. A 3-credit course (~135 hours) covers OS fundamentals. Use *Operating System Concepts* by Silberschatz et al. and **MIT's Operating System Engineering** course. Implement mini operating systems or kernel modules for practical understanding.

4. Networks & Internet

4.1 Computer Networking Fundamentals

Definition. A **computer network** is a system that connects multiple devices to transmit and share information [4]. It typically uses cables or wireless media and can include devices in cloud environments [20].

Key subtopics.

- **Network types:** LAN, WAN, WLAN, MAN, peer-to-peer vs. client-server architectures [21].
- **OSI and TCP/IP models:** layers (physical, data link, network, transport, application) and protocols at each layer (Ethernet, IP, TCP, UDP, HTTP, DNS, DHCP, FTP, SMTP etc.) [22].
- **IP addressing and subnetting.** IPv4/v6, routing, switching.
- **Network hardware:** routers, switches, modems, access points.
- **Network security:** firewalls, VPNs, intrusion detection systems.
- **Performance concepts:** bandwidth, latency, throughput, Quality of Service (QoS).

Learning path. A typical networking course is 3 credits (~135 hours). Recommended resources: **Kurose & Ross'** ***Computer Networking: A Top-Down Approach***, **Cisco's CCNA curriculum**, and **Andrew Tanenbaum's** ***Computer Networks***. Hands-on labs using network simulators (Packet Tracer, GNS3) help reinforce concepts.

4.2 Web Development

Topics.

- **Front-end:** HTML, CSS and JavaScript; frameworks/libraries like React, Angular or Vue; responsive design principles.
- **Back-end:** Server-side languages (Node.js, Python, Java, PHP), RESTful APIs, GraphQL, authentication and authorization.

- **Databases:** Integration with SQL/NoSQL databases.
- **Web protocols:** HTTP/HTTPS, WebSockets, TLS/SSL.
- **Full stack development:** combining front-end and back-end into deployable applications.

Learning path. Two semesters (6 credits). Start with HTML/CSS/JS, then server-side programming and frameworks. Projects should include building and deploying full-stack applications.

5. Databases & Data Management

Definition. A **database management system (DBMS)** is software for creating and managing databases; it enables end-users to create, protect, read, update and delete data and ensures data integrity and concurrency [5].

Key subtopics.

- **Data models:** relational, hierarchical, network, object-oriented.
- **Structured Query Language (SQL):** SELECT, INSERT, UPDATE, DELETE, joins, aggregation.
- **Schema design & normalization.** Entity-relationship modeling, normal forms.
- **Transaction management:** ACID properties, concurrency control, locking, isolation levels.
- **Indexing & performance tuning.** B-trees, hashing, query optimization.
- **NoSQL databases:** key-value stores, document stores, columnar databases, graph databases.
- **Data warehousing & big data:** OLAP vs. OLTP, data lakes, distributed storage (HDFS, Amazon S3).

Learning path. A 3-credit course (~135 hours). Use *Database System Concepts* by Silberschatz et al., *SQL for Data Analysis* resources and **Stanford's Databases** course. Practical labs on MySQL/PostgreSQL and NoSQL systems (MongoDB) are crucial.

6. Cybersecurity

Definition. Cybersecurity refers to **protecting systems, networks, devices and data from digital threats such as unauthorized access, cyberattacks and data breaches** [23]. It aims to safeguard the confidentiality, integrity and availability (CIA triad) of information [6].

Key subtopics.

- **Cryptography:** symmetric/asymmetric encryption, hashing, digital signatures, key exchange, TLS/SSL.
- **Network security:** firewalls, intrusion detection and prevention systems, secure protocols (HTTPS, SSH, VPNs).
- **Application security:** secure coding, input validation, authentication, session management, penetration testing.
- **Operating system & host security:** user privileges, patching, antivirus, endpoint protection.
- **Risk management & compliance:** threat modeling, vulnerability assessment, security policies, incident response, NIST Cybersecurity Framework.
- **Social engineering & human factors.**

Learning path. A typical cybersecurity fundamentals course is 3 credits (~135 hours). Additional specialized courses may cover cryptography, ethical hacking or secure software development. Suggested resources: **Coursera's *Introduction to Cyber Security* by NYU**, *Security Engineering* by Ross Anderson, **CISSP exam materials**.

7. Cloud Computing & Virtualization

Cloud computing definition. The National Institute of Standards and Technology defines cloud computing as a **model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service-provider interaction** [7]. Essential characteristics include on-demand self-service, broad network access, resource pooling, rapid elasticity and measured service [24].

Virtualization definition. Virtualization involves **creating a virtual version of a physical technology**—an operating system, server, storage device or network resource—using software to simulate hardware functionality [8]. It enables multiple operating systems and applications to run on a single physical machine [25].

Key subtopics.

- **Service models:** Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS).
- **Deployment models:** public, private, hybrid and community clouds.
- **Virtualization technologies:** hypervisors (Type 1 and Type 2), virtual machines, containers (Docker, containerd) and orchestration (Kubernetes).
- **Serverless computing & Function as a Service (FaaS).**
- **Cloud storage & databases:** object storage (S3, Azure Blob Storage), block storage, database as a service.
- **Security & compliance in the cloud.**
- **Monitoring & cost optimization.**

Learning path. A 3-credit course (~135 hours). Combine theoretical study with hands-on labs using cloud providers (AWS, Azure, GCP). Recommended resources: **AWS Certified Cloud Practitioner** training, **Google Cloud Fundamentals** and the NIST SP 800-145 document [7].

8. Data Science & Statistics

Definition. Data science is **a field of study that uses scientific methods, processes and systems to extract knowledge and insights from data** [9]. It is interdisciplinary, combining statistics, computer science and domain expertise [26].

Key subtopics.

- **Statistical foundations:** descriptive statistics, probability theory, hypothesis testing, regression, Bayesian methods.
- **Data wrangling & cleaning:** handling missing data, transformations, feature engineering.
- **Exploratory data analysis (EDA):** visualization, summary statistics.
- **Machine learning basics:** supervised, unsupervised and reinforcement learning (see section 9).
- **Big data tools:** Hadoop, Spark, distributed computing.
- **Data visualization:** Matplotlib, Seaborn, Tableau.
- **Ethical considerations:** data privacy, fairness and bias.

Learning path. Two courses (~2 × 3 credits). Start with a statistics course, then a data science/machine learning course. Use resources like **Harvard's Data Science Professional Certificate**, *Python for Data Analysis* by Wes McKinney and **Coursera's *Data Science* specialization by Johns Hopkins**.

9. Machine Learning & Artificial Intelligence

Machine learning definition. Machine learning is a **subfield of artificial intelligence that gives computers the ability to learn without being explicitly programmed** [10]. AI is broadly defined as the capability of a machine to imitate intelligent human behavior [27]. Machine learning techniques build models that can make predictions or decisions based on data [28].

Key subtopics.

- **Supervised learning:** regression, classification, support vector machines, decision trees, random forests, gradient boosting.
- **Unsupervised learning:** clustering (k-means, hierarchical), dimensionality reduction (PCA, t-SNE), anomaly detection.
- **Reinforcement learning:** agents, rewards, value functions, Q-learning.
- **Neural networks & deep learning:** perceptrons, multilayer neural networks, convolutional neural networks (CNNs) for image processing, recurrent neural networks (RNNs) for sequential data, transformers for language processing.
- **Model evaluation & tuning:** cross-validation, overfitting/underfitting, hyperparameter tuning.
- **Deployment & MLOps.**

Learning path. Two courses (~2 × 3 credits). Start with an introductory machine learning course (e.g., Andrew Ng's *Machine Learning* on Coursera). Then study deep learning (through *DeepLearning.AI* or MIT's *Deep Learning for Self-Driving Cars*). Practical projects (building models in Python with scikit-learn, TensorFlow or PyTorch) are essential.

10. Blockchain & Decentralized Systems

Definition. A **blockchain** is a collaborative, tamper-resistant ledger that maintains transactional records grouped into blocks; each block includes a unique identifier from the previous block, so changing data in one block alters subsequent blocks and provides tamper evidence [11].

Key subtopics.

- **Distributed ledger technology:** peer-to-peer networks, nodes, decentralization.
- **Consensus algorithms:** Proof of Work (PoW), Proof of Stake (PoS), Practical Byzantine Fault Tolerance (PBFT).
- **Cryptography:** hashing, Merkle trees, digital signatures.
- **Smart contracts:** programmable agreements (e.g., Ethereum's Solidity language).
- **Public vs. private blockchains:** Bitcoin, Ethereum, Hyperledger.
- **Decentralized applications (dApps).**
- **Challenges & limitations:** scalability, energy consumption, security vulnerabilities.

Learning path. A short course (~8–12 weeks). Study cryptographic foundations, then build simple smart contracts. Resources include **Coursera's *Blockchain Basics***, **IBM's *Blockchain for Developers* course** and the **Ethereum Developer documentation**.

11. Internet of Things (IoT)

Definition. The **internet of things (IoT)** is a network of interrelated devices that connect and exchange data with other IoT devices and the cloud [12]. IoT devices are embedded with technology such as

sensors and software; data transfer over the network does not require human-to-human or human-to-computer interactions [29].

Key subtopics.

- **Embedded systems & microcontrollers:** sensors, actuators, microprocessors (Arduino, Raspberry Pi).
- **Connectivity:** Wi-Fi, Bluetooth, Zigbee, LoRaWAN, NB-IoT.
- **Protocols & standards:** MQTT, CoAP, HTTP, MQTT-SN.
- **Edge computing & gateways:** processing data locally to reduce latency and bandwidth [30].
- **Security & privacy:** authentication, encryption, secure provisioning and firmware updates.
- **Applications:** smart homes, industrial IoT, healthcare, agriculture.

Learning path. A specialized course (~8–12 weeks). Begin with electronics and sensors, then network communication protocols. Build simple IoT projects (e.g., sensor networks) and use cloud IoT platforms (AWS IoT Core, Azure IoT Hub). Refer to **Cisco's *Introduction to IoT*** and **resources from the IoT Consortium**.

12. Virtualization & Containerization

Definition. Virtualization is the **creation of a virtual version of an actual piece of technology**, such as an OS, server, storage device or network resource [8]. It allows multiple operating systems and applications to run on a single server, increasing efficiency [25].

Key subtopics.

- **Hypervisors:** Type 1 (bare-metal) and Type 2 (hosted) hypervisors; examples include VMware ESXi, KVM, Hyper-V.
- **Virtual machines vs. containers:** differences, use cases and performance trade-offs.
- **Containerization:** Docker images, container registries, container networking.
- **Orchestration:** Kubernetes, Docker Swarm, Nomad.
- **Serverless computing:** functions executed on demand (AWS Lambda, Azure Functions).

Learning path. Typically combined with cloud computing courses. Aim for one semester (3 credits) to cover virtualization, containers and orchestration. Practical labs using virtualization software and container orchestration tools are essential.

13. Human-Computer Interaction (HCI) & User Experience Design

Definition. User Experience (UX) refers to the **feeling users experience when using a product, application, system or service** [13]. It encompasses the holistic journey users traverse, including their direct interactions and how the product fits into their overall task [31]. Human-Computer Interaction (HCI) studies how people interact with technology to design usable and accessible systems.

Key subtopics.

- **UX design principles:** usability, accessibility, consistency, feedback, error tolerance.
- **User research & personas:** surveys, interviews, observations, empathy mapping.
- **Information architecture & wireframing.**
- **Interaction design & prototyping:** low-fidelity and high-fidelity prototypes, iterative testing.
- **Accessibility standards (WCAG), inclusive design.**

- **Design tools:** Figma, Adobe XD, Sketch.
- **Evaluation methods:** usability testing, heuristic evaluation, A/B testing.

Learning path. A course (~8–12 weeks). Practice by designing and prototyping interfaces, conducting usability studies and iterating based on feedback. Resources: **Interaction Design Foundation's courses**, *Don't Make Me Think* by Steve Krug, and **Stanford's HCI course**.

14. Immersive Technologies: Augmented Reality (AR) & Virtual Reality (VR)

Definitions.

- **Augmented Reality (AR)** overlays digital information onto the real world, typically through a smartphone or tablet camera [32].
- **Virtual Reality (VR)** creates a completely artificial environment that immerses the user [33].

Key subtopics.

- **AR frameworks:** ARKit (Apple), ARCore (Google), WebXR.
- **VR platforms:** Oculus (Meta Quest), HTC Vive, PlayStation VR.
- **3D graphics & rendering:** coordinate systems, shaders, game engines (Unity, Unreal).
- **Interaction techniques:** gaze, gesture, controllers, haptics.
- **Spatial computing & mapping.**
- **Applications:** gaming, training, education, architecture, healthcare.
- **Challenges:** motion sickness, user safety, privacy, cost [34].

Learning path. A focused course (~6–8 weeks). Begin with AR/VR concepts and frameworks, then build simple immersive applications. Resources: **Coursera's *Introduction to Augmented Reality and ARCore***, **Unity Learn** and **MIT Media Lab lectures**.

15. Ethics, Privacy & Law in Technology

As technology pervades society, ethical considerations and legal frameworks become critical.

Topics.

- **Data privacy laws:** GDPR, CCPA, India's DPDP Act; consent, data minimization and user rights.
- **Intellectual property:** copyrights, patents, open-source licenses.
- **Bias & fairness:** algorithmic bias, discrimination, transparency and accountability in AI systems.
- **Cyberlaw & regulation:** digital rights, liability, jurisdiction for online crimes.
- **Ethical hacking & responsible disclosure.**
- **Sustainability & environmental impacts of technology.**

Learning path. A short course (~6–8 weeks). Combine reading of regulations with case studies. Resources: **EDX's *Ethics and Law in Data and Technology***, **Stanford's *Ethics in AI***, and guidance from organizations like the **Electronic Frontier Foundation (EFF)**.

Using This Map

1. **Survey broadly.** Start with the high-level overview table to understand the scope of technology domains.
2. **Identify interests.** Pick domains that align with your goals (e.g., programming, cybersecurity, data science).

3. **Follow recommended learning paths.** Each section provides approximate time commitments and resources. Adjust pace based on your background.
4. **Engage in projects.** Theory is essential but practical application cements understanding. Build small projects or contribute to open-source repositories.
5. **Stay current.** Technology evolves rapidly; subscribe to reputable news sources, join communities and practice lifelong learning.

Conclusion

This guide cannot capture literally “the entire internet,” but it provides a structured map of the major technological domains that underpin modern computing and the web. By starting with high-level overviews and then diving into specific subfields, you can systematically acquire deep expertise. The credit-hour–based time estimates [1] help plan your learning journey like a professional university curriculum. With curiosity, consistent practice and the resources listed, you can explore technology to whatever depth you desire.

---

[1] Credit Hour Definition | University of Cincinnati
https://www.uc.edu/about/registrar/registration/policies/credit-hour-definition.html

[2] Computer Organization and Architecture Tutorial - GeeksforGeeks
https://www.geeksforgeeks.org/computer-organization-architecture/computer-organization-and-architecture-tutorials/

[3] [18] [19] Introduction to Operating System - GeeksforGeeks
https://www.geeksforgeeks.org/operating-systems/introduction-of-operating-system-set-1/

[4] [20] [21] [22] What is a Computer Network? - Florida National University (FNU)
https://www.fnu.edu/what-is-a-computer-network/

[5] What is a Database Management System (DBMS)? | Definition from TechTarget
https://www.techtarget.com/searchdatamanagement/definition/database-management-system

[6] [23] What Is Cybersecurity? An In-Depth Guide | Park University
https://www.park.edu/blog/what-is-cybersecurity-an-introduction/

[7] [24] NIST SP 800-145, The NIST Definition of Cloud Computing
https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-145.pdf

[8] [25] What Is Virtualization? | Definition from TechTarget
https://www.techtarget.com/searchitoperations/definition/virtualization

[9] [26] What is data science
https://seas.harvard.edu/news/what-data-science-definition-skills-applications-more

[10] [27] [28] Machine learning, explained | MIT Sloan
https://mitsloan.mit.edu/ideas-made-to-matter/machine-learning-explained

[11] Blockchain | NIST
https://www.nist.gov/blockchain

[12] [29] [30] What is IoT (Internet of Things)? | Definition from TechTarget
https://www.techtarget.com/iotagenda/definition/Internet-of-Things-IoT

[13] [31] What is User Experience? | Definition and Overview
https://www.productplan.com/glossary/user-experience/

[14] [32] [33] [34] Virtual Reality (VR) and Augmented Reality (AR) | The Princeton Review
https://www.princetonreview.com/ai-education/vr-and-ar

15 What is Data Structure? - GeeksforGeeks
https://www.geeksforgeeks.org/dsa/data-structure-meaning/

16 17 What is DevOps? | Atlassian
https://www.atlassian.com/devops