

DCE TECH CHRONICLE

— IDEAS | INNOVATION | TECHNOLOGY —

CSE EDITION

Volume VI, Edition II
(January - June 2026)



CONTENTS

01



HOD Message

02



Editor's Message

03



Vision & Mission

04



PEOs, POs, PSOs

05



Student's Articles

06



Department
Achievements



HOD's MESSAGE

Dr. Ashima Mehta
(HOD CSE)



“Technology does not simply change the world; it empowers us to reimagine what is possible.”

It is a matter of great pride to introduce this technical magazine, a reflection of the ideas, imagination, and innovative spirit of our Computer Science & Engineering community. In an era where Artificial Intelligence, Data Science, Cybersecurity, Cloud Computing, and emerging technologies are redefining the way we live and work, the ability to learn, question, create, and innovate has become more important than ever.

This magazine is more than a collection of technical articles—it is a glimpse into the curiosity and potential of our students. Every idea shared, project explored, and insight presented represents a step towards becoming not just technology users, but technology creators and problem-solvers.

As a department, our endeavour is to provide an environment where students can experiment with ideas, embrace challenges, learn from failures, and transform their knowledge into meaningful solutions. I encourage our students to remain curious, keep exploring beyond the classroom, and have the courage to turn their ideas into reality.

My appreciation goes to the faculty members, students, and editorial team whose dedication has brought this edition to life. I hope these pages spark new thoughts, inspire innovation, and encourage every reader to ask one important question: “What can I create that has never been created before?”

Wishing the CSE community a future filled with curiosity, creativity, and limitless possibilities.

Dr. Ashima Mehta
Head, Department of Computer Science & Engineering

Editor's Message



Editor in-Chief

It gives me immense pleasure to present this edition of our Technical Magazine. This platform showcases the creativity, curiosity and technical spirit of our students and faculty. I hope it inspires every reader to learn, explore, innovate and create.

Prof. Vimmi Malhotra
(Faculty; CSE Department)

STUDENT EDITORS



Swayam Prabha

CO Editor

Roll No.: 26184; 3rd year



Veer Bhadra Singh

Editor (Design)

Roll No.: 26195; 3rd year



Kanishka

Editor (Text)

Roll No.: 26081; 3rd year

Small Idea, Big Impact



Vision

Steering the future of computer science through innovative advancements, fostering ethical values and principles through technical education.



Mission

- Directing future innovations in computer science through revolutionary progress.
- Instilling a foundation of ethical values and principles in every technologist.
- Offering a comprehensive technical education to equip individuals for a meaningful and influential future.

Program Educational Objectives (PEO)

- Apply the technical competence in Computer Science and Engineering for solving problems in the real world.
- Carry out research and develop solutions on problems of social applications.
- Work in a corporate environment, demonstrating team skills, work morals, flexibility and lifelong learning.

Program Specific Outcome (PSO)

- Exhibit design and programming skills to develop and mechanize business solutions using revolutionary technologies.
- Learn strong theoretical foundation leading to brilliance and enthusiasm towards research, to provide well-designed solutions to complicated problems.
- Work effectively with diverse Engineering fields as a team to design, build and develop system applications.

Programme Outcome (Pos)

- **PO1:** Engineering Knowledge: Apply knowledge of mathematics, natural science, computing, engineering fundamentals and an engineering specialization as specified in WK1 to WK4 respectively to develop to the solution of complex engineering problems.
- **PO2:** Problem Analysis: Identify, formulate, review research literature and analyze complex engineering problems reaching substantiated conclusions with consideration for sustainable development. (WK1 to WK4)
- **PO3:** Design/Development of Solutions: Design creative solutions for complex engineering problems and design/develop systems/components/processes to meet identified needs with consideration for the public health and safety, whole-life cost, net zero carbon, culture, society and environment as required. (WK5)
- **PO4:** Conduct Investigations of Complex Problems: Conduct investigations of complex engineering problems using research-based knowledge including design of experiments, modelling, analysis & interpretation of data to provide valid conclusions. (WK8).
- **PO5:** Engineering Tool Usage: Create, select and apply appropriate techniques, resources and modern engineering & IT tools, including prediction and modelling recognizing their limitations to solve complex engineering problems. (WK2 and WK6)
- **PO6:** The Engineer and The World: Analyze and evaluate societal and environmental aspects while solving complex engineering problems for its impact on sustainability with reference to economy, health, safety, legal framework, culture and environment. (WK1, WK5, and WK7).
- **PO7:** Ethics: Apply ethical principles and commit to professional ethics, human values, diversity and inclusion; adhere to national & international laws. (WK9)
- **PO8:** Individual and Collaborative Team work: Function effectively as an individual, and as a member or leader in diverse/multi-disciplinary teams.
- **PO9:** Communication: Communicate effectively and inclusively within the engineering community and society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations considering cultural, language, and learning differences
- **PO10:** Project Management and Finance: Apply knowledge and understanding of engineering management principles and economic decision-making and apply these to one's own work, as a member and leader in a team, and to manage projects and in multidisciplinary environments.
- **PO11:** Life-Long Learning: Recognize the need for, and have the preparation and ability for i) independent and life-long learning ii) adaptability to new and emerging technologies and iii) critical thinking in the broadest context of technological change.(WK8)

Student's Articles

DevSecOps: Making Security Part of Software Development



Jigyasu Gaur
Roll. No. : 25074(Final Year)

Modern software is no longer developed once and released after months of testing. Applications are continuously updated, new features are added frequently, and software is often deployed across cloud platforms and distributed environments. While this approach has made software delivery faster, it has also created new opportunities for security vulnerabilities to enter the system.

Traditionally, security testing was often performed near the end of the Software Development Life Cycle (SDLC). This approach can create a serious problem: a vulnerability discovered after deployment may require developers to revisit code, redesign components, repeat testing, and sometimes interrupt services. The cost and effort of fixing such problems can therefore increase significantly.

DevSecOps addresses this challenge by making security a continuous part of software development and delivery. Instead of treating security as a final checkpoint, DevSecOps integrates security practices into development, testing, deployment and monitoring.

From DevOps to DevSecOps

Student's Articles

DevOps brought development and operations teams closer together with the goal of delivering software rapidly and reliably. Automation, continuous integration and continuous delivery became important parts of this approach.

However, faster development without adequate security can create another problem. A vulnerability can move through an automated pipeline just as quickly as a legitimate feature.

DevSecOps extends the DevOps philosophy by introducing security into the same workflow. Developers, operations teams and security professionals work together to identify and address security issues throughout the development process.



The fundamental idea is simple:

Security should not be a final inspection, it should be part of the development process itself.

Student's Articles

Security Across the Software Development Lifecycle

A DevSecOps pipeline can integrate security activities at multiple stages.

Plan: Security requirements and potential risks are considered while defining the application.

Code: Developers follow secure coding practices and use code-review mechanisms to identify potentially unsafe implementations.

Build: Automated tools can inspect source code and software dependencies for known vulnerabilities.

Test: Applications can undergo different forms of security testing before deployment.

Deploy: Security configurations, access controls and deployment environments are checked before software reaches production.

Monitor: After deployment, applications and infrastructure are continuously monitored for suspicious activity, vulnerabilities and unexpected behaviour.



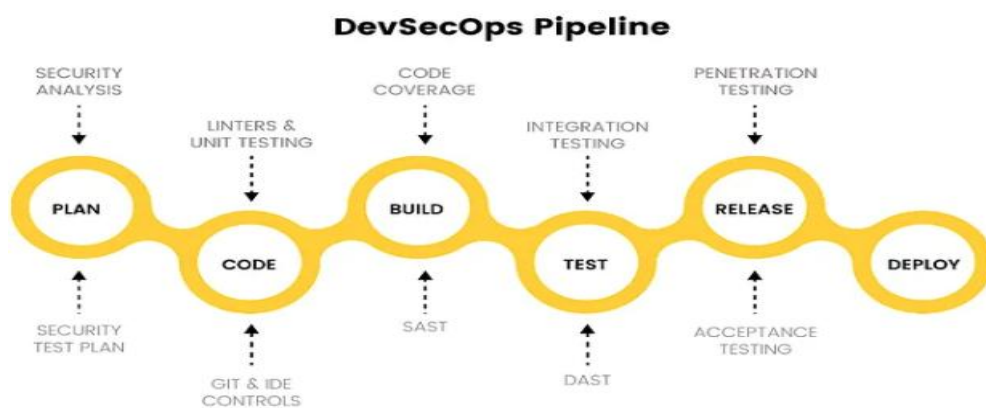
This approach is described as “**shifting security left.**” The objective is to identify problems as early as possible, when they are generally easier to understand and fix.

What Happens Inside a DevSecOps Pipeline?

Consider a web application being developed by a software team. A developer modifies the application and pushes the updated code into a version-control system. This action can automatically trigger a CI/CD pipeline

Student's Articles

. Automated tools can then examine the source code, The pipeline may first compile or build the application. dependencies and configuration files for security weaknesses. If a serious vulnerability is detected, the pipeline can stop the release and notify the development team. If the security checks are successfully completed, the application can proceed toward deployment. Once deployed, monitoring systems continue looking for abnormal behaviour or newly discovered vulnerabilities.



Therefore, security becomes a **continuous process rather than a single event**.

Important Security Practices

One of the important components of DevSecOps is **automated security testing**. Automated tools can examine applications repeatedly without requiring every security check to be performed manually.

Static Application Security Testing (SAST) examines source code or compiled code to identify potentially insecure programming practices. It can help developers discover problems during development.

Student's Articles

Dynamic Application Security Testing (DAST) examines a running application from the outside and attempts to identify vulnerabilities that may become visible during execution. Modern applications rarely depend entirely on code written by their own developers. They often use open-source libraries and external packages. A vulnerable dependency can therefore introduce a security weakness even when the application's own code appears safe. DevSecOps practices can also include container security, secrets management, access control, infrastructure security and continuous vulnerability monitoring.

Why DevSecOps Matters

The major advantage of DevSecOps is that it brings security closer to the point where software is created.

Finding a security problem during development allows developers to understand the code responsible for the issue and correct it before the application reaches users.

Automated checks can also reduce the possibility of human error and make security testing more consistent.

DevSecOps can also encourage a **shared responsibility for security**. Security is no longer viewed as the exclusive responsibility of a separate security team. Developers, testers, system administrators and security professionals all contribute to protecting the application.

Student's Articles

Challenges in Adopting DevSecOps

DevSecOps is not simply a matter of installing security tools. Its successful implementation requires changes in both technology and organisational culture.

Developers need sufficient knowledge of secure coding practices. Security tools may sometimes generate false positives, requiring teams to distinguish genuine vulnerabilities from harmless findings. Automated security checks can also increase pipeline execution time if they are poorly designed. Another challenge is balancing **development speed with security requirements**. Organisations must avoid creating security processes so complicated that they slow development unnecessarily.

Training, automation, clearly defined responsibilities and appropriate security policies are therefore essential for successful DevSecOps adoption.

DevSecOps Challenges



People
(culture and training)



Adapting security
for modern cloud
environments



Technically complex



**Lack of
DevOps talent**



Expensive



Operations
security misalignment



Not enough time



**Risks across
the pipeline**

Student's Articles

The Future of Secure Software Development

As software becomes increasingly dependent on cloud platforms, APIs, containers, open-source components and automated deployment systems, security will become even more closely connected with software engineering.

Future development environments are likely to make security checks increasingly automated and intelligent. Developers may receive security feedback while writing code rather than discovering vulnerabilities after deployment.

However, technology alone cannot guarantee secure software. Human judgement remains important for understanding business risks, evaluating vulnerabilities and deciding how security should be implemented.

In Conclusion, DevSecOps represents a change in the way software security is approached. Instead of asking **“Is the software secure before release?”**, development teams begin asking **“How can security be maintained throughout the entire software lifecycle?”**

By integrating security into planning, coding, testing, deployment and monitoring, DevSecOps helps organisations identify vulnerabilities earlier and build security into the software rather than attempting to add it later.

For future software engineers, understanding DevSecOps is therefore not merely about learning another development methodology. It is about recognising that **good software must be functional, reliable and secure from the very beginning.**

Student's Articles

Open-Source AI: Opportunity, Innovation and Responsibility



Radhika Khatter

Roll. No. : 26144(3rd Year)

Artificial intelligence is no longer limited to large technology companies with massive computing infrastructure. The growth of open-source AI has made sophisticated AI technologies increasingly accessible to developers, students, researchers and smaller organisations. Instead of depending entirely on proprietary platforms, developers can explore, modify, integrate and build upon openly available AI technologies.

This shift has created an interesting question: **What happens when powerful AI becomes accessible to almost everyone?**

Open-source AI has the potential to accelerate innovation, encourage collaboration and reduce barriers to experimentation. At the same time, greater accessibility can introduce concerns related to security, misuse, privacy, licensing and responsible development

What Does Open-Source AI Mean?

The term *open-source AI* generally refers to AI software, models, tools or supporting components that are made available with permissions that allow people to use, study, modify or redistribute them, subject to the applicable licence. This is different from a completely closed AI system, where the underlying model, implementation or training resources may not be publicly accessible.

An open approach can allow developers to examine how a technology works, adapt it for particular applications and contribute improvements to the wider community. However, “open” does not automatically mean that every part of an AI system is freely available. Different projects can make different components accessible under different licences.

Student's Articles



For developers, understanding the **licence and usage conditions** is therefore just as important as understanding the technology itself.

Why Is Open-Source AI Important?

One of the greatest strengths of open-source AI is **accessibility**. Students and developers can experiment with AI technologies without having to build an entire system from the ground up.

Open development can also encourage collaboration. A developer working on one aspect of an AI system may improve it, while another developer may adapt that improvement for a different application. Over time, a community can collectively develop tools that would be difficult for an individual developer to create alone. This ecosystem provides an opportunity to move beyond simply *using* AI applications. They can study software structures, experiment with models, build interfaces, optimise applications and understand how AI technologies can be integrated into larger software systems.

Student's Articles

Open Source and Software Innovation

Open-source development has historically played an important role in software engineering. Operating systems, programming tools, databases, web frameworks and countless development libraries have benefited from collaborative development. Open-source AI extends this philosophy into an area where software, data and computational resources interact. For example, a developer could take an available AI model and build a specialised application around it rather than developing an entire AI system from scratch. Another developer might optimise the software for a particular hardware environment, while someone else could create a user interface or integration layer. This creates an **innovation chain** in which different developers contribute different capabilities. The result can be faster experimentation and greater diversity of applications.

The Responsibility That Comes With Accessibility

Greater access to technology also means greater responsibility. An AI model or software component released openly can potentially be used for beneficial purposes such as education, accessibility, software development and research. However, the same technology could potentially be misused to automate harmful activities, generate misleading content or create security threats.

This creates a difficult balance. Restricting access too heavily can reduce innovation and research opportunities, while unrestricted access can make misuse easier.

Responsible open-source AI therefore requires developers to think beyond the question of “**Can we build this?**” and consider “**How this technology could be used?**”

Security and Privacy Concerns

Open-source AI systems can introduce security concerns at several levels.

Student's Articles



A developer may unknowingly use an outdated model or software dependency containing a vulnerability. An improperly configured AI application could also expose sensitive information. When AI systems are integrated into larger applications, weaknesses in APIs, databases or authentication mechanisms can create additional risks. Developers must carefully examine what information enters an AI system, where that information is processed and whether sensitive data could be exposed.

An AI component should be treated as part of a complete software system, not as an isolated piece of technology.

The Importance of Licensing

Open-source projects operate under licences, and those licences can impose different conditions on how software or models may be used, modified or redistributed.

For developers, ignoring licensing requirements can create legal and technical problems. Before incorporating an open-source AI component into a project, developers should examine its licence, documentation and usage conditions.

This is particularly important when software is intended for commercial deployment.

Student's Articles

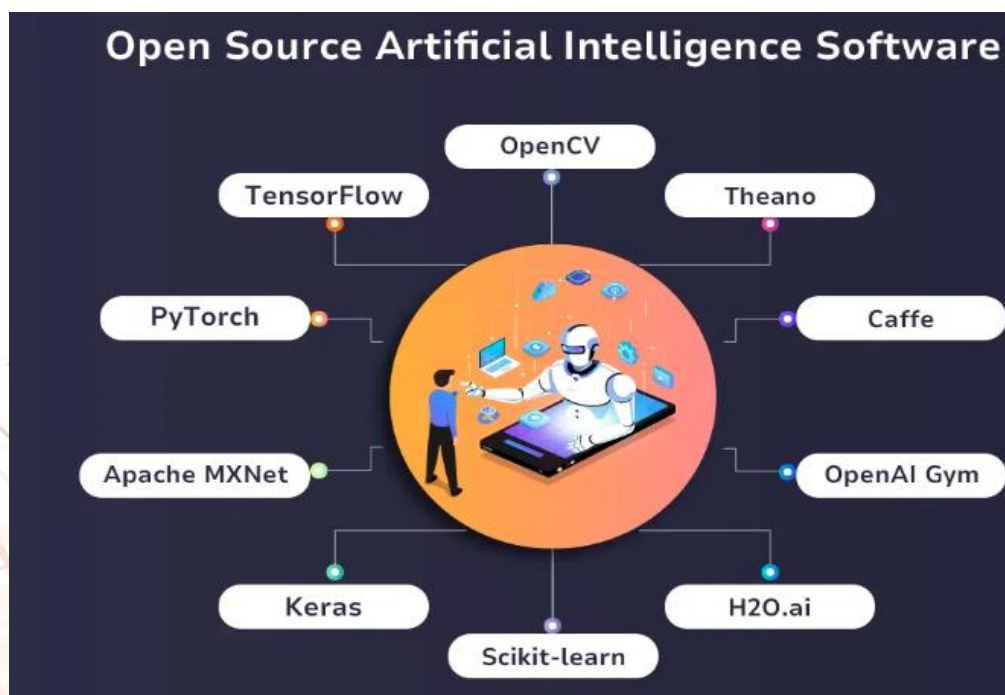
Therefore, responsible AI development requires not only programming knowledge but also an understanding of **software licensing and intellectual property**.

Open-Source AI in Education

Open-source AI can have a significant role in computer science education.

Students can experiment with AI systems, study their software components, develop applications and compare different approaches. Instead of treating AI as a black-box service, students can investigate how different software components interact within an application.

For example, a student project could involve developing a programming assistant, document-search application or educational chatbot using an openly available AI model. The important learning outcome is not simply producing the final application but understanding its architecture, limitations, security requirements and responsible usage.



Student's Articles

What Lies Ahead?

The future of open-source AI is likely to involve increasing interaction between AI models and conventional software systems. Developers may use AI components as building blocks alongside databases, APIs, cloud services and user interfaces. At the same time, communities will need better practices for evaluating models, documenting limitations, managing licences, reporting vulnerabilities and reducing misuse. Technical progress will therefore need to be accompanied by responsible engineering practices.

In Conclusion, Open-source AI represents more than the availability of AI models and tools. It represents a shift towards greater participation in AI development. Students, researchers, developers and organisations can experiment, collaborate and create new applications without relying entirely on closed platforms.

However, openness comes with responsibility. Security vulnerabilities, privacy risks, licensing requirements and potential misuse cannot be ignored.

The future of AI should not only be about making powerful technology accessible, it should also be about making its development thoughtful, secure and responsible.



Student's Articles

SQL vs NoSQL: Choosing the Right Database Architecture



Riddhi
Roll No.: 26153(3rd Year)

Almost every modern software application depends on data. Whether it is a college management system, an e-commerce platform, a banking application or a social networking service, applications need a reliable way to store, retrieve and manage information.

For many years, **relational databases** have been the standard choice for managing structured data. SQL-based systems provide tables, relationships, constraints and a well-defined structure. However, the rapid growth of web applications, cloud platforms and large-scale digital services has created situations where traditional database architectures may not always be the most convenient solution.

This has contributed to the growth of **NoSQL databases**, which provide alternative ways of storing and accessing data.

SQL vs. NoSQL



Student's Articles

The question, therefore, is not simply “*Which database is better?*” but rather:

“Which database architecture is appropriate for a particular application?”

Understanding SQL Databases

SQL, or Structured Query Language, is commonly associated with relational database management systems. Data is organised into tables containing rows and columns, and relationships can be established between different tables. Consider a college database containing information about students, courses and departments. Instead of storing everything in one large table, the information can be divided into related tables. A student record can be connected to a course record through defined relationships.

This structured approach provides consistency and makes relational databases particularly useful for applications where relationships between data are important.

SQL databases also support transactions, allowing a group of related operations to be treated as a single unit. This is especially valuable in applications such as banking, where an incomplete transaction can cause serious problems.

What Makes NoSQL Different?

NoSQL databases emerged to address some of the challenges associated with large-scale and rapidly changing applications. Unlike relational databases, NoSQL systems do not necessarily organise information into traditional tables and rows. Different NoSQL databases may use different data models, including **document, key-value, wide-column and graph-based structures.**

Student's Articles

A document database, for example, can store related information together in a flexible document structure. This can be useful when application data changes frequently or when developers need to work with information that does not fit neatly into fixed tables. NoSQL does not mean that such databases cannot work with structured data. Rather, it represents a broader family of database approaches designed for different requirements.

SQL vs NoSQL: A Practical Comparison

The distinction becomes clearer when we consider how applications use their data.

Factor	SQL Databases	NoSQL Databases
Data structure	Structured tables	Flexible models
Schema	Generally predefined	Often more flexible
Relationships	Strong support	Depends on database type
Transactions	Strong transactional support	Varies by system
Scaling	Traditionally strong vertical scaling; horizontal scaling is also possible	Often designed with horizontal scaling in mind
Best suited for	Structured, relationship-heavy applications	Flexible, large-scale or rapidly changing data
Query approach	SQL	Varies by database
Data consistency	Strong consistency is commonly supported	Depends on database and configuration

Student's Articles

This table should not be interpreted as a competition in which one technology always wins. Database selection depends heavily on the application's requirements.

When SQL Makes More Sense

SQL databases are often a strong choice when an application has **well-defined relationships and strict data consistency requirements**.

A banking system is a simple example. Suppose a customer transfers money from one account to another. The system must ensure that money is deducted from one account and added to the other correctly. A database transaction can help ensure that either the complete operation succeeds or the appropriate rollback occurs.

Similarly, university management systems, inventory systems and many enterprise applications contain structured information with relationships that benefit from relational database design. SQL is also powerful when applications require complex queries involving multiple related tables.

When NoSQL Can Be Useful

NoSQL databases can become attractive when applications handle very large volumes of data, require flexible data structures or need to scale across distributed infrastructure.

Consider a social media application. User profiles, posts, comments, reactions and other information can have different structures and may change over time. A flexible database model can simplify certain development scenarios.

Student's Articles

NoSQL databases can also be useful for applications involving rapidly generated data, large-scale web services, caching, real-time systems and distributed applications.

However, choosing NoSQL simply because an application is “large” is not enough. The data model, consistency requirements, query patterns and operational environment must all be considered.



The Role of Scalability

Scalability is one of the major factors behind the SQL–NoSQL discussion. Traditional database deployments have often relied heavily on **vertical scaling**, where additional computing resources are added to a server. Distributed NoSQL systems are frequently designed around **horizontal scaling**, where workloads can be distributed across multiple machines.

However, the distinction is not absolute. Modern relational databases can also support replication, partitioning, clustering and distributed architectures. Therefore, database architecture should be selected according to the application's expected workload rather than simply following the idea that “SQL is for small systems and NoSQL is for large systems.”

Student's Articles

Can SQL and NoSQL Work Together?

In real software systems, developers do not always have to choose one technology for everything.

An application may use a relational database for critical transactional information while using a NoSQL database for another type of workload. For example, structured customer or payment records could remain in an SQL database, while a separate system handles high-volume logs or flexible application data.

This approach is sometimes called **polyglot persistence**, where different data-storage technologies are selected according to different application requirements.

Challenges in Database Selection

Choosing a database architecture involves several considerations.

Developers must understand the expected data structure, query patterns, transaction requirements, scalability needs, security considerations and operational complexity. Moving from one database architecture to another later can also be expensive if the application has been designed too closely around a particular technology.

Another important consideration is the expertise available within the development team. A technically suitable database may still create difficulties if developers do not understand its data model and operational requirements.

Database selection is therefore an architectural decision, not simply a programming preference.

Student's Articles



In Conclusion, SQL and NoSQL represent different approaches to solving the same fundamental problem: **how to store and retrieve information effectively.**

SQL databases remain highly valuable for structured data, complex relationships and applications requiring strong transactional guarantees. NoSQL databases provide alternative models that can be particularly useful for flexible data structures and distributed, large-scale workloads.

The choice between them should therefore not be based on popularity or technology trends. A good software engineer first understands the application's requirements and then selects the database architecture that best satisfies those requirements.

Student's Articles

DIGITAL TWINS: BUILDING VIRTUAL VERSIONS OF THE REAL WORLD



Pranavi Gote
Roll No.: 25115(Final Year)

When the Physical World Gets a Digital Brain

Imagine a factory where every machine has a virtual counterpart. Before changing a machine setting in the real world, engineers can test it digitally. Imagine a smart city where traffic patterns, energy consumption, buildings, and public infrastructure can be monitored through a continuously updated virtual model. This is the idea behind **Digital Twin technology**.

A Digital Twin is a virtual representation of a physical or real-world entity, system, process, or environment. Unlike a simple 3D model, a digital twin can be connected to real-world data and updated throughout the lifecycle of the physical system. NIST describes digital twins as virtual representations that can support monitoring, simulation, prediction, optimization, and decision-making.



Student's Articles

When the Physical World Gets a Digital Brain

Imagine a factory where every machine has a virtual counterpart. Before changing a machine setting in the real world, engineers can test it digitally. Imagine a smart city where traffic patterns, energy consumption, buildings, and public infrastructure can be monitored through a continuously updated virtual model. This is the idea behind **Digital Twin technology**.

A Digital Twin is a virtual representation of a physical or real-world entity, system, process, or environment. Unlike a simple 3D model, a digital twin can be connected to real-world data and updated throughout the lifecycle of the physical system. NIST describes digital twins as virtual representations that can support monitoring, simulation, prediction, optimization, and decision-making.

WHAT EXACTLY IS A DIGITAL TWIN?

A **Digital Twin** is a **virtual counterpart of a physical object, system, or process** that is continuously updated using real-world data.

For example, sensors in a manufacturing machine can collect **temperature, vibration, pressure, speed, and energy data**. This information helps engineers monitor performance, detect unusual patterns, and **predict potential failures**.

Unlike a traditional computer model, a Digital Twin maintains a connection with the physical system, enabling **real-time monitoring, analysis, simulation, and two-way data exchange**.

Student's Articles

HOW DOES DIGITAL TWIN TECHNOLOGY WORK?

Digital Twins bring together several important areas of computer science and engineering:

1. Internet of Things (IoT)

Sensors and connected devices collect information from physical objects and environments.

2. Data Communication

The collected information is transmitted through networks to digital systems for processing and analysis.

3. Data Storage & Cloud Computing

Large quantities of sensor and historical data can be stored and processed using cloud or edge computing infrastructure.

4. Artificial Intelligence & Machine Learning

AI and machine-learning algorithms can identify patterns, detect anomalies, and support predictions.

5. Modelling & Simulation

Computer models simulate how a physical system behaves under different conditions.

6. Visualization

Dashboards, 3D environments, graphs, and augmented or virtual reality can help users understand complex systems.

Together, these technologies transform raw sensor data into actionable information. NIST identifies sensors, Industrial Internet of Things infrastructure, simulation models, and data-driven synchronization as important elements of effective Digital Twin systems.

DIGITAL TWINS IN MANUFACTURING

Digital Twins help manufacturers create virtual replicas of machines, production lines, and processes. Engineers can test

Student's Articles

operations, monitor equipment, and predict potential failures before they occur. By analysing real-time data such as vibration and temperature, Digital Twins support **predictive maintenance, reduced downtime, improved efficiency, and smarter production planning.**

DIGITAL TWINS IN SMART CITIES

The concept becomes even more powerful when applied to entire cities.

A Smart City Digital Twin can combine information from traffic sensors, buildings, energy systems, transportation networks, environmental sensors, and other connected infrastructure.

Planners can use the digital environment to explore possible changes before implementing them in the physical city.

For example:

What happens if a new road is constructed?

How will traffic change during peak hours?

Where charging stations should be installed?

How might energy consumption change?

Digital simulation can help planners examine such scenarios before making expensive real-world decisions. Digital Twin platforms can model environments ranging from individual buildings and factories to energy networks, railways, stadiums, and entire cities.

APPLICATIONS BEYOND FACTORIES

Digital Twin technology is transforming multiple industries beyond manufacturing and smart cities:

Healthcare: Supports patient modelling, medical research, and hospital operations.

Student's Articles

Automotive: Enables virtual vehicle design, testing, and performance analysis.

Energy: Monitors power systems, equipment, and energy networks for better efficiency.

Buildings: Tracks energy use, equipment, occupancy, and environmental conditions.

Transportation: Models roads, railways, and traffic systems to improve planning and mobility.



DIGITAL TWIN VS. SIMULATION

Digital Twins and simulations both use computer-based models, but they serve different purposes.

A **Simulation** models a system to test different scenarios and understand “**What could happen?**”

A **Digital Twin** is connected to a real-world system and uses its data to monitor performance, identify issues, and predict “**What is happening and what could happen next?**”

In short:

Simulation → Tests possibilities

Digital Twin → Monitors, predicts, and supports decisions

Student's Articles

BENEFITS OF DIGITAL TWINS

Digital Twin technology can provide several important advantages:

Predictive Maintenance: Identify potential equipment failures before they become major problems.

Reduced Costs: Test changes virtually before implementing expensive physical modifications.

Improved Efficiency: Optimize processes, machines, energy consumption, and resources.

Better Decision-Making: Use real-time and historical information to support decisions.

Risk Reduction: Test potentially risky scenarios in a virtual environment.

Faster Innovation: Evaluate new designs and processes digitally.

Lifecycle Management: Track systems from design and development through operation and maintenance.

The ultimate goal is not simply to create a beautiful digital model. The value comes from using data, models, and analysis to make better decisions.

CHALLENGES AND CYBER SECURITY

Digital Twins rely on large amounts of accurate data, making **data quality, system integration, privacy, and cyber security** key challenges. Unauthorized access to a connected Digital Twin could potentially affect real-world systems. Strong security, trusted data, and common standards are therefore essential.

THE FUTURE OF DIGITAL TWINS

Student's Articles

The future of Digital Twins lies in combining **AI, IoT, Edge Computing, robotics, extended reality, and advanced simulation**. AI can make twins more predictive, while edge computing enables faster real-time analysis. As these technologies advance, Digital Twins could evolve into intelligent systems that **monitor, predict, and support real-world decisions**.

CONCLUSION

Digital Twins bridge the **physical and digital worlds** by combining IoT, AI, cloud computing, data analytics, and simulation. They help industries **monitor systems, predict problems, test solutions, and make smarter decisions** across manufacturing, healthcare, transportation, energy, and smart cities.

For CSE, Digital Twins bring together **AI, machine learning, cloud computing, cyber security, networking, databases, and software engineering**, making them an important technology for the future.

**PHYSICAL WORLD + DATA + AI + SIMULATION =
INTELLIGENT DIGITAL TWIN**



Department's Achievements

ACHIEVERS OF GATE-2026



GATE Score:
871(CS)

All India Rank
130

Mr. Harsh Monga
Roll No : 25058

GATE Registration No :
CS26S43012039

Mark Obtained : **74.58**



GATE Score:
731(CS)

All India Rank
801

Mr. Aryan Prakhar
Roll No : 24020

GATE Registration No :
CS26S34038070

Mark Obtained : **62.6**



GATE Score:
730(CS)

All India Rank
819

Mr. Akash Chauhan
Roll No : 24001

GATE Registration No :
CS26S44023222

Mark Obtained : **62.5**



GATE Score:
711(DA)

All India Rank
330

Mr. Piyush Garg
Roll No : 22109

GATE Registration No :
DA26S83044121

Mark Obtained : **56.33**



GATE Score:
705(CS)

All India Rank
1107

Mr. Piyush Garg
Roll No : 22109

GATE Registration No :
CS26S33044124

Mark Obtained : **60.32**



GATE Score:
625(CS)

All India Rank
2519

Mr. Akshat Sharma
Roll No : 26021

GATE Registration No :
CS26S33044138

Mark Obtained : **53.48**



GATE Score:
602(DA)

All India Rank
1030

Mr. Aryan Prakhar
Roll No : 24020

GATE Registration No :
DA26S84038077

Mark Obtained : **47.33**



GATE Score:
586(CS)

All India Rank
3536

Mr. Kanak Tanwar
Roll No : 25077

GATE Registration No :
CS26S33062161

Mark Obtained : **50.22**



GATE Score:
556(CS)

All India Rank
4545

Ms. Barkha Yadav
Roll No : 26039

GATE Registration No :
CS26S43009111

Mark Obtained : **47.65**



GATE Score:
546(DA)

All India Rank
1718

Mr. Harsh Monga
Roll No : 25058

GATE Registration No :
DA26S83012085

Mark Obtained : **42.67**



GATE Score:
532(CS)

All India Rank
5711

Mr. Chandan Yadav
Roll No : 22039

GATE Registration No :
CS26S34038070

Mark Obtained : **45.58**



GATE Score:
529(CS)

All India Rank
5820

Mr. Parth Sharma
Roll No : 26130

GATE Registration No :
CS26S33062133

Mark Obtained : **45.34**



GATE Score:
510(CS)

All India Rank
6956

Ms. Riya
Roll No : 25058

GATE Registration No :
CS26S33009126

Mark Obtained : **43.71**



GATE Score:
486(DA)

All India Rank
3024

Mr. Akshat Sharma
Roll No : 26021

GATE Registration No :
DA26S83044135

Mark Obtained : **37.67**



GATE Score:
480(CS)

All India Rank
8959

Mr. Saksham Bhatnagar
Roll No : 25159

GATE Registration No :
CS26S33053113

Mark Obtained : **41.1**

Department's Achievements

ACHIEVERS OF GATE-2026



GATE Score:
449(CS)

All India Rank
11941

Ms. Ipshita Rawat
Roll No. : 26075

GATE Registration No :
CS26S33050038

Mark Obtained : **38.5**



GATE Score:
447(CS)

All India Rank
12126

Ms. Yogita Kanyal
Roll No. : 25203

GATE Registration No :
CS26S43055051

Mark Obtained : **38.33**



GATE Score:
443(CS)

All India Rank
12527

Mr. Pranav Joshi
Roll No. : 25117

GATE Registration No :
CS26S43044154

Mark Obtained : **37.99**



GATE Score:
438(CS)

All India Rank
13243

Mr. Yash Kumar
Roll No. : 25207

GATE Registration No :
CS26S33060245

Mark Obtained : **37.52**



GATE Score:
431(CS)

All India Rank
13875

Mr. Pranav Kandpal
Roll No. : 26136

GATE Registration No :
CS26S43036145

Mark Obtained : **36.95**



GATE Score:
430(DA)

All India Rank
4886

Mr. Harsh Jadaun
Roll No. : 26067

GATE Registration No :
DA26S83041118

Mark Obtained : **33**



GATE Score:
430(DA)

All India Rank
4886

Mr. Saksham Bhatnagar
Roll No. : 25159

GATE Registration No :
DA26S83053118

Mark Obtained : **33**



GATE Score:
430(CS)

All India Rank
14175

Mr. Harshit Hasija
Roll No. : 25059

GATE Registration No :
CS26S33043123

Mark Obtained : **36.87**



GATE Score:
426(CS)

All India Rank
14658

Mr. Aman Dhingra
Roll No. : 26024

GATE Registration No :
CS26S33043024

Mark Obtained : **36.54**



GATE Score:
419(CS)

All India Rank
15384

Mr. Gourav Dhankhar
Roll No. : 24401

GATE Registration No :
CS26S43041104

Mark Obtained : **35.92**



GATE Score:
414(DA)

All India Rank
5690

Ms. Manya Sehgal
Roll No. : 25091

GATE Registration No :
DA26S83009026

Mark Obtained : **31.67**



GATE Score:
411(CS)

All India Rank
16511

Mr. Sarthak Gumber
Roll No. : 26166

GATE Registration No :
CS26S33060143

Mark Obtained : **35.24**



GATE Score:
411(CS)

All India Rank
16511

Ms. Kanishka
Roll No. : 26081

GATE Registration No :
CS26S33009108

Mark Obtained : **35.24**



GATE Score:
411(CS)

All India Rank
16810

Ms. Garima Ratra
Roll No. : 26058

GATE Registration No :
CS26S43009061

Mark Obtained : **35.22**



GATE Score:
405(DA)

All India Rank
6110

Ms. Yogita Kanyal
Roll No. : 25203

GATE Registration No :
DA26S83055085

Mark Obtained : **31**

Department's Achievements

ACHIEVERS OF GATE-2026



GATE Score:
405(DA)

All India Rank
6110

Mr. Ashish Kumar Jha
Roll No : 24051

GATE Registration No :
DA26S83043077

Mark Obtained : **31**



GATE Score:
403(CS)

All India Rank
18020

Ms. Chhavi Luthra
Roll No : 26048

GATE Registration No :
CS26S43034100

Mark Obtained : **34.53**



GATE Score:
403(CS)

All India Rank
18020

Mr. Veer Bhadra Singh
Roll No : 26195

GATE Registration No :
CS26S43043126

Mark Obtained : **34.53**



GATE Score:
385(DA)

All India Rank
7275

Ms. Dipti
Roll No : 25046

GATE Registration No :
DA26S83053099

Mark Obtained : **29.33**



GATE Score:
385(CS)

All India Rank
21041

Mr. Anuj
Roll No : 26028

GATE Registration No :
CS26S33062054

Mark Obtained : **32.96**



GATE Score:
383(CS)

All India Rank
21411

Ms. Radhika Khatter
Roll No : 24020

GATE Registration No :
CS26S43041144

Mark Obtained : **32.81**



GATE Score:
379(CS)

All India Rank
22192

Mr. Aneesh Trisal
Roll No : 25013

GATE Registration No :
CS26S43059269

Mark Obtained : **32.46**



GATE Score:
369(DA)

All India Rank
8364

Ms. Riya
Roll No : 26158

GATE Registration No :
DA26S83009104

Mark Obtained : **28**



GATE Score:
362(CS)

All India Rank
26076

Ms. Chahat Yadav
Roll No : 26047

GATE Registration No :
CS26S33041200

Mark Obtained : **31**



GATE Score:
362(CS)

All India Rank
26076

Mr. Amit
Roll No : 26022

GATE Registration No :
CS26S33012202

Mark Obtained : **31**



GATE Score:
362(CS)

All India Rank
26076

Ms. Gunik Chhabra
Roll No : 26064

GATE Registration No :
CS26S33009111

Mark Obtained : **31**



GATE Score:
359(CS)

All India Rank
26568

Mr. Ayush Gupta
Roll No : 26037

GATE Registration No :
CS26S43043127

Mark Obtained : **30.74**



GATE Score:
358(CS)

All India Rank
27094

Mr. Ashish Kumar Jha
Roll No : 24051

GATE Registration No :
CS26S33043074

Mark Obtained : **30.68**



GATE Score:
357(DA)

All India Rank
9277

Mr. Aryan Bhasin
Roll No : 26032

GATE Registration No :
DA26S83044134

Mark Obtained : **27**



GATE Score:
354(CS)

All India Rank
28114

Mr. Aayush Bhardwaj
Roll No : 26004

GATE Registration No :
CS26S33061295

Mark Obtained : **30.35**

Department's Achievements

ACHIEVERS OF GATE-2026



GATE Score:
346(CS)

All India Rank
29612

Ms. Yogita Yadav
Roll No : 26202

GATE Registration No:
CS26S33010203

Mark Obtained : **29.7**



GATE Score:
338(CS)

All India Rank
32388

Mr. Puru Gupta
Roll No : 25131

GATE Registration No:
CS26S43043101

Mark Obtained : **29.01**



GATE Score:
334(CS)

All India Rank
33516

Mr. Adarsh Kumar
Roll No : 26012

GATE Registration No:
CS26S43061277

Mark Obtained : **28.67**



GATE Score:
333(DA)

All India Rank
11297

Mr. Jiten Baldia
Roll No : 25075

GATE Registration No:
DA26S83043130

Mark Obtained : **25**



GATE Score:
320(CS)

All India Rank
37938

Ms. Riddhi
Roll No : 26153

GATE Registration No:
CS26S33061003

Mark Obtained : **27.42**



GATE Score:
318(CS)

All India Rank
38620

Ms. Minakshi
Roll No : 24913

GATE Registration No:
CS26S48025061

Mark Obtained : **27.28**



GATE Score:
318(CS)

All India Rank
38620

Mr. Debajyoti Ray
Roll No : 23043

GATE Registration No:
CS26S43055099

Mark Obtained : **27.28**



GATE Score:
316(CS)

All India Rank
39301

Mr. Sujal Roy
Roll No : 26181

GATE Registration No:
CS26S33060095

Mark Obtained : **27.1**



GATE Score:
298(CS)

All India Rank
46071

Mr. Saurav Kumar
Roll No : 24144

GATE Registration No:
CS26S43041059

Mark Obtained : **25.26**



GATE Score:
450(DA)

All India Rank
4134

Mr. Aum Singhal
Roll No : 23029

GATE Registration No:
DA26S82059323

Mark Obtained : **34.67**

Department's Achievements

ACHIEVERS OF GATE-2026



DRONACHARYA
College of Engineering



Congratulations!

For Qualifying **GATE 2026**

Graduate Aptitude Test in Engineering

AIR

8074



GATE Score : 492

Prof. Renu Narwal

Assistant Professor (CSE)

Mark Obtained : **42.13**



Exemplary Success in GATE 2026

Prof. Renu Narwal achieved a commendable GATE 2026 score of 492, securing AIR 8074. Her accomplishment reflects dedication and academic excellence, inspiring the entire institution.