BBDNIIT

techno inspire@IT

An Annual Magazine from Department of Information Technology

Vol. 5, Issue I, 2024

Department of Information Technology

Babu Banarasi Das

Northern India Institute of Technology

Chief Patrons

Mrs. Alka Das
Hon'ble Chairperson, BBD Educational Group

Mr. Viraj Sagar Das
Hon'ble President, BBD Educational Group

Ms. Sonakshi Das
Hon'ble Vice President, BBD Educational Group

Patrons

Mr. R. K.Agarwal
Chief Executive Director, BBDEG

Prof (Dr.) Shailendra Kumar Saraf
Director (Pharmacy), BBDNIIT

Prof (Dr.) V. K. Singh
Director (Engg.), BBDNIIT

Chief-Editor

Prof (Dr.) Bhawesh Kumar Thakur
Professor & Head,
Department of Information Technology, BBDNIIT, Lucknow

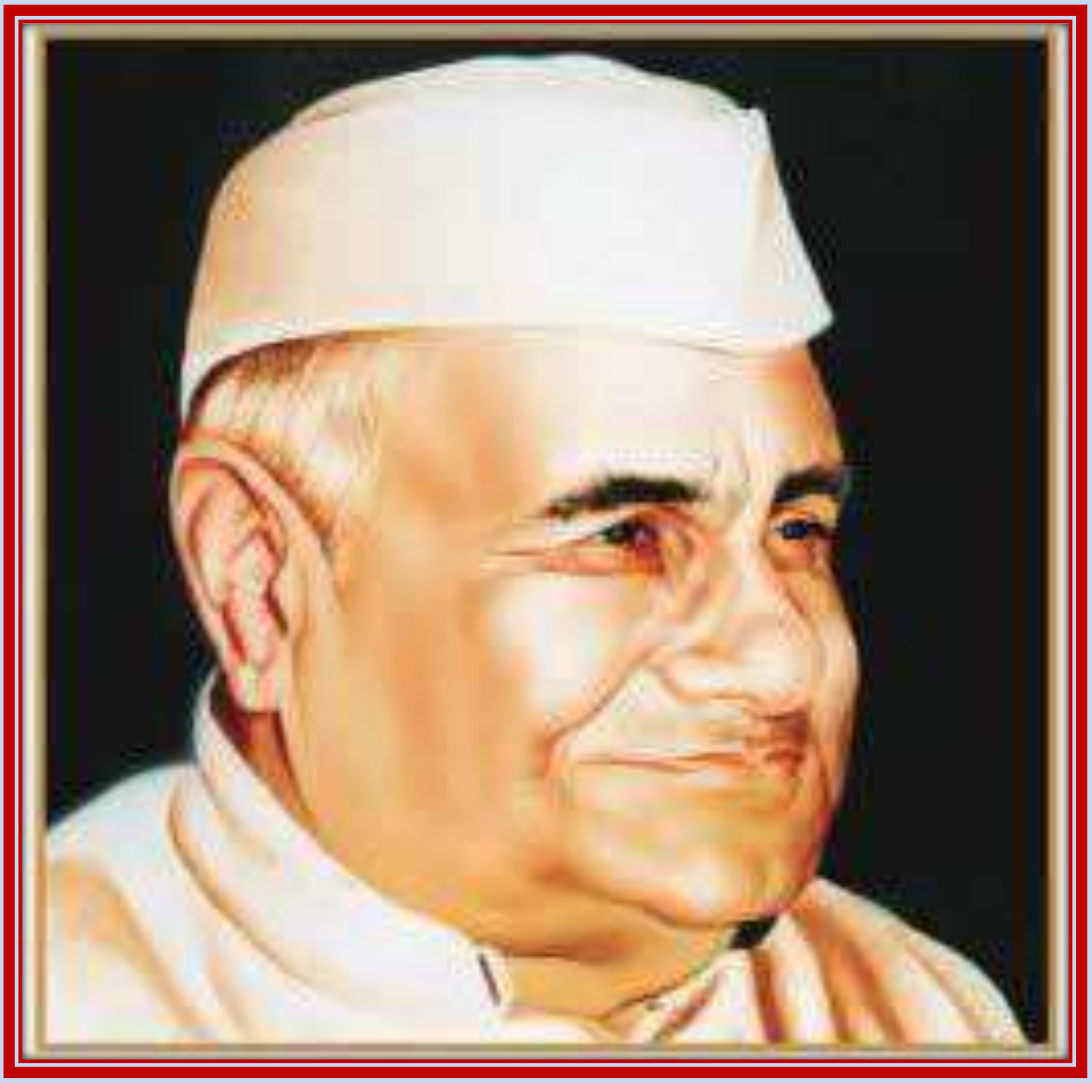
Faculty Editors

- **Mr. Praveen Sachan,** Assistant Professor, IT
- **Ms. Neelam Chakravarti,** Assistant Professor, IT
- **Mr. Reetesh Gupta,** Assistant Professor, IT
- **Ms. Shaba Irram,** Assistant Professor, IT
- **Ms. Priya Sharma,** Assistant Professor, IT

Student Editors

- **Mr. Siddharth Raj,** IT, Fourth Year
- **Ms. Shruti Saxena,** IT, Fourth Year
- **Mr. Utkarsh Joshi,** IT, Third Year
- **Ms. Ishita Singh,** IT, Third Year
- **Mr. Utkarsh Kumar Srivastava,** IT, Second Year
- **Ms. Awani Shukla,** IT, Second Year

Genesis



Late Babu Banarasi Das Ji
(1912 – 1985)
Ex Chief Minister, Govt. of Uttar Pradesh

"To provide an open opportunity to the young generation for evolving their core competencies and to build their career as world class professionals with broad based foundation, in-depth knowledge & versatile personality to meet the challenges of Global Economy…" – Babuji's Vision

Hon'ble Founder

Late Dr. Akhilesh Das Gupta

(1961 – 2017)

Ex-Member of Parliament (Rajya Sabha)

Hon'ble Founder & Ex-Chairman, BBD Group

"We not only make technocrats at BBD, we churn out citizens of the world, perfect in all respect, be it leadership, competence, confidence, communication, moral or knowledge…" – *Dr. Akhilesh Das Gupta's Vision*

A distinguished political leader, philanthropist, educationist and social worker, Dr. Akhilesh Das is the pride of Lucknow city. He has established himself as a worthy son of his illustrious father Late Babu Banarasi Das Ji.

Public Life

He has obtained MBA, LLB & Ph.D. (Human Resource Management) degrees; Dr. Akhilesh Das decided to follow his father's footprints by choosing Social Service as his primary goal. In his distinguished public life spanning well over three decades, Dr. Das was first elected as the Mayor of Lucknow in year 1993

and became one of the youngest Mayors of the world. Subsequently, he was elected as a ***Member of Parliament (Rajya Sabha)*** *in November 1996 and was re-elected for a second term in November 2002 till November 2008. Dr. Das has been elected again as a Member of Parliament (Rajya Sabha) for a third consecutive term in 2008 till 2014.*

Dr. Das has been a member of many important parliamentary committees and National level apex advisory & statutory bodies including the AIMS. He has been a member of the Executive Committee of AICTE. He had been the Chairman Parliamentary Standing Committee for Industries. He also served with distinction as the Union Minister of State for Steel and made a place for himself as one of the foremost National leaders.

Contribution to Sports

Dr. Akhilesh Das was an enthusiastic sports person. He underwent an exhaustive training in England as a Junior Badminton player. Later, he represented and captained the U.P. State Badminton team for many years and won many National & International tournaments including the National Doubles Championship. His love for other sports is no less. He has been instrumental in organizing and facilitating many National level tournaments in various sports. He was the president of the U.P. Olympic Association & Chairman of Badminton Association of India.

Contribution to the Cause of Education

*As a social thinker, Dr. Das realized the need of Quality Technical & Professional education for the overall growth and upliftment of the masses and development of the State of Uttar Pradesh. Accordingly, Babu Banarasi Das Educational Society (****BBDES****) was founded through which the torch for higher professional education was lit in Lucknow, the Capital of U.P. The first institution in the series, Babu Banarasi Das Institute of Technology & Management (****BBDITM****), Lucknow was established in 1998 and Babu Banarasi Das Northern Indian Institute of Technology (****BBDNIIT****), Lucknow in 1999, Dr. Akhilesh Das Gupta Institute of Technology & Management (****ADGITM****), New Delhi in 2003, Babu Banarasi Das Engineering College (****BBDEC****), Lucknow in 2009 and Babu Banarasi Das University (****BBDU****), Lucknow in 2011.*

Hon'ble Chairperson, BBD Group

Mrs. Alka Das

A lady with spirit, determination and zeal, Mrs. Alka Das has been the driving force behind her husband Dr. Akhilesh Das. She is the daughter of Late Justice D. P. Gupta, who held high positions in the State of Rajasthan, as its Governor and Chief Justice. She was brought up on Gandhian ideals and philanthropic traditions.

Mrs. Das is a Management expert and a legal luminary who has a long involvement with issues of social justice and women empowerment in our country. She is a legal consultant to several reputed organizations and an active Social Worker. She is connected with many National Social & Educational Organizations. She is also a Life Member of the "Rajasthan High Court Bar Association". The establishment of Northern India Engineering College, Lucknow was an initiative taken by her to provide quality professional education to match the International Standards.

As a Co-founder of Babu Banarasi Das Group of Educational Institutions, Mrs. Das has made a fertile attempt to nurture a generation of self-made and highly motivated students with an inbuilt commitment to do something constructive for the Country.

Hon'ble President, BBD Group

Mr. Viraj Sagar Das

Vice- President, Badminton Association of India
President, Uttar Pradesh Olympic Association

"Education is the most powerful weapon which you can use to change the world" *– Nelson Mandela*

Babu Banarasi Das Educational Group has its roots lying deep in quality education practices that equip each student with broad outlooks to prepare them for this futuristic world. We are committed to providing the education that fulfils the needs of today's world. Today, education is not limited to textbook courses and its practical implementation. Education has been through a major transformation where apart from the syllabus, it has become imperative to inculcate some extra skills in the students so that they could truly succeed in their lives. We encourage our students to perform well in their academics and also to take active participation in all the co-curricular activities in order to develop interpersonal skills, take up challenges, and be self-dependent and most importantly, have a good time while learning.

Vision of the Institute

"To establish a multi-disciplinary environment with excellence in technical education and research for developing competent professionals who meet the challenges of industrial and societal development with human values and ethics."

Mission of the Institute

M1. *To provide an excellent environment with supporting infrastructure to prepare globally competent professionals acceptable to industry and society*

M2. *To inculcate a spirit of research, innovation and entrepreneurship by exposing multi-disciplinary approach*

M3. *To motivate aspiring graduates to solve real life problems with zeal of lifelong learning*

M4. *To imbibe a healthy environment which helps to develop intellectual capabilities among graduates to transform them into professionals with human values and ethics*

About the Department

The Department of Information Technology offers Bachelor's degree in Information Technology. This degree program prepares graduates for successful, profitable and lifelong careers in IT and ITES fields. Information Technology students study software systems through innovative classroom instructions, supported by laboratories equipped with the state of-the-art hardware and software. The department ensures that the students are introduced to both fundamental and advanced knowledge in areas such as Networking technology, Computer Security and software engineering, Cloud Computing, Data Analytics etc.

Salient Features of the department are as follows.

- State of the Art Computing Facilities
- ICT in Teaching and Learning
- Teacher as Mentor
- Research & Publications with Social Impact
- Supportive Learning for Placements
- Professional Development for Industrial Engagement

Various activities conducted by the department for the students are as follows.

- Value Added Courses
- Career Oriented Add-On Courses
- Special Training for Competitive Examination
- Expert Lectures for Industry Interaction
- Report Writing & Paper Presentation
- Personality Development & Soft Skill Training

Vision of the Department

"To be a dynamic and premier center for imparting transformative education with quality teaching-learning to excel students in the field of Information Technology and to meet the expectations of the industry and society with moral values and ethics."

Mission of the Department

M1: To provide cutting edge knowledge based education through compatible infrastructure to develop professionals who cater the need of Industry and society at large.

M2: To motivate graduates for achieving excellence in academics, industry and entrepreneurship to serve the society.

M3: To inspire students for transformation in industry and society with their noble values and professional ethics.

Program Educational Objectives (PEOs)

The Program Educational Objectives of the U.G. program in the Department of Information Technology are:

After 4 years of completing this program, the students:

PEO-1: Graduates will be able to apply procedural and programmatic knowledge with innovative skills to provide solutions in a technology-driven era.

PEO-2: Graduates will possess professional competency to be employable, entrepreneur and inclination towards higher education with a zeal for life-long learning.

PEO-3: Graduates will be a valuable asset for society by exhibiting their professional engineering quality with moral values and ethics.

Program Specific Objectives (PSOs)

PSO-1: The graduates should have ability to provide engineering solutions in the field of Image Processing, Machine Learning, Cloud Computing and Data Sciences using available IT tools and techniques.

PSO-2: The graduates should have ability to employ inter-disciplinary approaches to design and develop IT solutions for real world problems.

Program Outcomes (POs)

Information Technology (IT) Graduates will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and Sustainability:** Understand the impact of the professional engineering solutions in societal and environmental context and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and Team Work:** Function effectively as an individual and as a member or leader in diverse teams and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with the society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project Management and Finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life Long Learning:** Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadcast of technological change.

Message from the Director (Engg.)

It gives me immense pleasure to present **Volume 5, Issue I (2024)** of the **Annual Magazine of the Department of Information Technology, BBDNIIT, Lucknow**. This publication reflects the creativity, enthusiasm, and intellectual spirit of our faculty members and students, who have voluntarily contributed their ideas, experiences, achievements, and creative expressions. Such initiatives foster a culture of knowledge sharing, innovation, and collaboration while providing a meaningful platform to showcase the diverse talents within the department.

I congratulate the editorial team and all the faculty and student contributors for their valuable efforts in bringing out this edition. I am confident that this magazine will inspire our students to explore emerging technologies, think creatively, and pursue excellence in academics, research, and professional development. I extend my best wishes to the Department of Information Technology for continued growth, innovation, and success.

Regards
Prof. (Dr.) V. K. Singh
Director (Engg.)
Babu Banarasi Das Northern India Institute of Technology, Lucknow

Message from Head of Department

It gives me immense pleasure to present **Volume 5, Issue I (2024)** of the **Annual Magazine of the Department of Information Technology, BBDNIIT, Lucknow**. This edition reflects the academic achievements, technical activities, creative pursuits, and vibrant spirit of our department. It provides a glimpse into the continuous efforts of our faculty and students towards learning, innovation, research, and professional development.

I am particularly delighted to acknowledge the voluntary contributions of our faculty members and students, whose ideas, experiences, technical insights, and creative expressions have enriched this publication. Their enthusiasm and willingness to share knowledge exemplify the collaborative spirit of our department and make this magazine a meaningful platform for communication and inspiration.

I congratulate the editorial team and all contributors for their sincere efforts in bringing out this edition. I hope this magazine encourages our students and faculty to explore new possibilities, embrace emerging technologies, and continue striving for excellence. My best wishes to the entire Department of Information Technology for continued growth and success.

Regards
Prof. (Dr.) Bhawesh Kumar Thakur
Head, Department of Information Technology
Babu Banarasi Das Northern India Institute of Technology, Lucknow

Faculty Contribution

Students Contribution

Faculty Contribution

Protecting Digital Identity in the AI Era

Mr. Yogesh Sharma
Assistant Professor, Department of Information Technology

Introduction
In today's connected world, our identity exists not only in the physical world but also across a vast digital ecosystem. Email accounts, social-media profiles, banking applications, educational portals, cloud services, Aadhaar-linked services, online shopping accounts and professional platforms collectively form our **digital identity**.
Digital identity includes the information and credentials used to recognize and authenticate an individual online. Traditionally, protecting this identity meant using strong passwords, antivirus software and basic security practices. However, the rapid development of Artificial Intelligence (AI) is changing the cybersecurity landscape. AI can strengthen identity protection, but it can also enable attackers to create highly convincing phishing messages, deepfakes, impersonation attacks and automated social engineering.
In April 2026, India's Computer Emergency Response Team (CERT-In) warned about frontier-AI-driven cyber risks, including AI-generated phishing, impersonation, automated credential harvesting and multi-stage attacks. It specifically advised individuals to strengthen authentication, remain alert to deepfakes and verify suspicious requests. [1] Therefore, protecting digital identity in the AI era requires more than simply remembering a strong password; it requires a combination of **technology, awareness, verification and responsible digital behaviour**.

Understanding Digital Identity
Digital identity is the collection of attributes, credentials and digital records associated with an individual or organization. It may include:

• Username and passwords • Email addresses and phone numbers • Government-issued digital identities • Biometric information • Banking and financial credentials	• Social-media profiles • Academic and professional records • Device and browser information • Digital certificates and cryptographic keys • Online behavioural information

A compromised identity can provide attackers with access to multiple services. If the same password is reused across platforms, compromising one account may lead to a chain of further compromises.
For organizations, identity is equally important. Employees, administrators, customers, applications, devices and services all require controlled access to digital resources. Consequently, **identity and access management (IAM)** has become a fundamental component of modern cybersecurity.

How AI Is Changing Identity Threats
AI is creating a new generation of cyber threats because attackers can automate activities that previously required considerable human effort.

AI-Powered Phishing
Traditional phishing messages often contain obvious spelling mistakes or suspicious formatting. Generative AI can produce grammatically correct, personalized and convincing messages in multiple languages.
An attacker could potentially analyze publicly available information about a person and create a message that appears to come from a colleague, teacher, manager or financial institution.
CERT-In has highlighted the increasing use of AI-generated phishing and social-engineering content as an emerging risk. [1]

Deepfakes and Impersonation
Deepfake technology can generate or manipulate realistic audio, video and images. A criminal could use a person's publicly available photograph or voice sample to create convincing fake content.
CERT-In's advisory on deepfakes identifies risks including misinformation, fraud, social engineering, financial loss and reputational damage. [2]
The most important lesson is that **seeing or hearing is no longer always sufficient evidence of authenticity**.

If someone sends a video message requesting an urgent financial transfer, or a voice call appears to come from a senior official, the recipient should independently verify the request.

Automated Credential Attacks
AI can help attackers analyze leaked credentials, identify potential targets and automate reconnaissance. India's CERT-In has warned that advanced AI systems may lower the barrier for attackers and increase the speed and scale of cyber campaigns. [1]
This makes password reuse particularly dangerous.

Why Passwords Alone Are No Longer Enough
Passwords remain one of the most common methods of authentication, but they have significant weaknesses. Passwords can be stolen through phishing, exposed through data breaches, guessed through brute-force attacks or reused across multiple services.

A better approach is **Multi-Factor Authentication (MFA)**, which requires two or more authentication factors, such as:

1. Something you know — password or PIN
2. Something you have — phone, security key or authenticator
3. Something you are — fingerprint, face or other biometric characteristic

CISA recommends phishing-resistant MFA, including security-key and passkey-based approaches, as stronger defenses against credential-based attacks. [3]
For organizations, MFA should be especially important for administrator and privileged accounts because compromise of such accounts can provide attackers with extensive access.

Passkeys: A More Secure Future
One promising development in digital identity is the **passkey**.
Passkeys use public-key cryptography rather than requiring users to remember traditional passwords. They can be protected using device biometrics, PINs or other local authentication mechanisms.
The security advantage is significant: the private cryptographic key remains on the user's device, while the service stores a corresponding public key. This reduces exposure to traditional password theft and phishing.
For IT professionals, understanding technologies such as **FIDO2, WebAuthn and passkeys** will become increasingly important as organizations move toward passwordless authentication.

AI as a Defender
Although AI creates new threats, it can also become a powerful cybersecurity defense mechanism.
AI-based security systems can analyze large volumes of activity and identify unusual behaviour. For example, an AI system may detect:

- Login attempts from unusual locations
- Abnormal access times
- Unusual device behaviour
- Multiple failed authentication attempts
- Suspicious account activity
- Unexpected data downloads
- Patterns associated with phishing or malware

This supports the concept of **behaviour-based identity protection**, where systems evaluate not only who a user claims to be but also whether the user's behaviour is consistent with normal activity.
However, AI-based security systems themselves must be carefully designed. NIST's AI Risk Management Framework emphasizes managing risks throughout the AI lifecycle and promoting trustworthy and responsible AI. [4]

Zero Trust: Never Assume, Always Verify
Traditional security models often assumed that users inside an organization's network could be trusted. Modern environments are much more complex because employees may work remotely, applications may run in the cloud, and devices may connect from different locations.
The **Zero Trust** approach challenges the assumption of automatic trust. Every access request should be evaluated based on factors such as identity, device, location, application, risk and requested resource.
The principle can be summarized as:
"Never trust automatically; verify continuously."

This approach is particularly relevant in the AI era because compromised credentials may look legitimate. A stolen username and password should not automatically provide unrestricted access.

Protecting Digital Identity: A Practical Strategy
Digital identity protection requires multiple layers of defense.

1. Use Unique and Strong Passwords
Every important account should have a unique password. A password manager can help generate and store strong credentials.

2. Enable MFA
MFA should be enabled wherever available, especially for email, banking, cloud, social-media and administrative accounts.

3. Prefer Phishing-Resistant Authentication
Where supported, use passkeys or hardware security keys rather than relying only on passwords or SMS-based authentication. CISA recommends stronger phishing-resistant methods where feasible. [3]

4. Verify Unexpected Requests
Do not immediately trust messages asking for passwords, OTPs, financial transfers or confidential information.
If a request appears urgent, independently contact the person using a trusted communication channel.

5. Be Careful with Voice and Video
AI-generated voices and videos can be convincing. A familiar voice should not automatically be treated as proof of identity.

6. Minimize Personal Information Online
The more personal information an individual publishes publicly, the easier it may become for attackers to construct convincing impersonation attempts.
Avoid unnecessarily sharing phone numbers, addresses, identification documents, travel plans and sensitive personal information.

7. Secure Devices
Keep operating systems, browsers, applications and security software updated. Device security is an essential part of identity security because an attacker who gains control of a device may obtain access to authenticated sessions and stored credentials.

8. Monitor Accounts
Regularly review login activity, connected devices, account recovery options and security notifications.

9. Treat AI Tools Carefully
Users should avoid entering confidential personal, financial, institutional or organizational information into unapproved AI services. CERT-In has also advised users to exercise caution with sensitive information when using AI applications. [5]

Digital Identity in the Indian Context
India's rapid digital transformation makes identity protection particularly significant. Digital services increasingly connect citizens with banking, education, healthcare, government services and commerce.
The **Digital Personal Data Protection Act, 2023** establishes a legal framework concerning the processing of digital personal data while recognizing individuals' rights over their personal data. [6] The Government of India also notified the **Digital Personal Data Protection Rules, 2025**, establishing further implementation requirements and an enforcement framework. [7]
For IT professionals, privacy should therefore be considered alongside cybersecurity during system design. Applications should collect only necessary information, protect sensitive data appropriately and provide mechanisms for responsible data management.

The Role of IT Professionals
IT professionals have a particularly important responsibility in protecting digital identity because they design and maintain the systems through which identities are created, authenticated and managed.

Developers should follow **secure-by-design** principles. Security should not be treated as an additional feature added after application development.
Important practices include:

• Secure authentication and session management • Strong access-control mechanisms • Role-Based Access Control (RBAC) • Encryption of sensitive information • Secure API design • Protection against credential attacks	• Logging and monitoring • Regular vulnerability assessment • Secure software development practices • Privacy-aware system design • Incident response planning

NIST's Cybersecurity Framework 2.0 provides a broad structure for organizations to manage cybersecurity risk through governance and security outcomes. [8]

Digital Identity Literacy for Students
Students of Information Technology are future developers, system administrators, cybersecurity professionals, entrepreneurs and technology leaders. They should therefore develop strong **digital identity literacy**.
Students should understand not only how to build authentication systems but also how those systems can fail.
A student developing a web application should ask:
Can an attacker steal a user's session?
Can a user access another user's data?
Can an attacker reset an account without proper verification?
Is MFA implemented securely?
Are passwords stored safely?
Can an AI-generated message fool the authentication process?
Thinking like both a developer and an attacker helps create stronger systems.

Conclusion
The AI era is fundamentally changing the meaning of digital identity. Artificial Intelligence can strengthen cybersecurity through intelligent monitoring, anomaly detection and automated defense, but the same technology can also make phishing, impersonation, deepfakes and social engineering more convincing and scalable.
Therefore, protecting digital identity requires a **defense-in-depth approach** combining strong authentication, MFA, passkeys, privacy protection, secure software development, continuous monitoring and human awareness.
The most important lesson is simple:

Do not trust a digital identity merely because it looks authentic. Verify it. Protect it. Monitor it.
For today's IT students and professionals, cybersecurity is no longer a specialized concern limited to security departments. Every application, device, database and AI system interacts with identity in some way. Building a safer digital future will therefore require technology professionals who understand that **security and privacy are not barriers to innovation—they are essential foundations of trustworthy innovation**.

References
[1] Indian Computer Emergency Response Team (CERT-In), "Defending Against Frontier AI Driven Cyber Risks," Advisory CIAD-2026-0020, April 26, 2026.
[2] Indian Computer Emergency Response Team (CERT-In), "Deepfakes – Threats and Countermeasures," Advisory CIAD-2024-0060, November 27, 2024.
[3] Cybersecurity and Infrastructure Security Agency (CISA), "Phishing Guidance: Stopping the Attack Cycle at Phase One," 2025; see also CISA guidance on phishing-resistant MFA.
[4] C. Autio et al., *Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile*, NIST AI 600-1, National Institute of Standards and Technology, 2024.
[5] Indian Computer Emergency Response Team (CERT-In), "Security Implications of AI Language Based Applications," Advisory, Government of India.
[6] Ministry of Electronics and Information Technology, Government of India, *Digital Personal Data Protection Act, 2023*, 2023.
[7] Ministry of Electronics and Information Technology, Government of India, *Digital Personal Data Protection Rules, 2025*, November 2025.
[8] C. Pascoe, S. Quinn and K. Scarfone, *The NIST Cybersecurity Framework (CSF) 2.0*, NIST Cybersecurity White Paper 29, National Institute of Standards and Technology, 2024.

Building Technology Startups from College Projects: From Classroom Ideas to Real-World Impact

Mr. Praveen Sachan
Assistant Professor, Department of Information Technology

Introduction

Every year, thousands of engineering and Information Technology students develop projects as part of their academic curriculum. These projects may include mobile applications, Artificial Intelligence (AI) models, Internet of Things (IoT) systems, cybersecurity solutions, healthcare applications, cloud platforms or data analytics systems. Traditionally, many projects are submitted for evaluation and then remain unused after the semester ends.

But a college project can be much more than an academic requirement. With the right approach, it can become a **prototype, product, business idea and potentially a technology startup**.

The journey requires a change in mindset—from *"How do we complete this project?"* to *"What real problem are we solving, who needs our solution, and how can we create sustainable value?"*

The Startup India initiative recognizes innovation and entrepreneurship as important components of India's startup ecosystem and provides support mechanisms for eligible startups [1].

From Project to Startup

The transformation can be understood through the following journey:

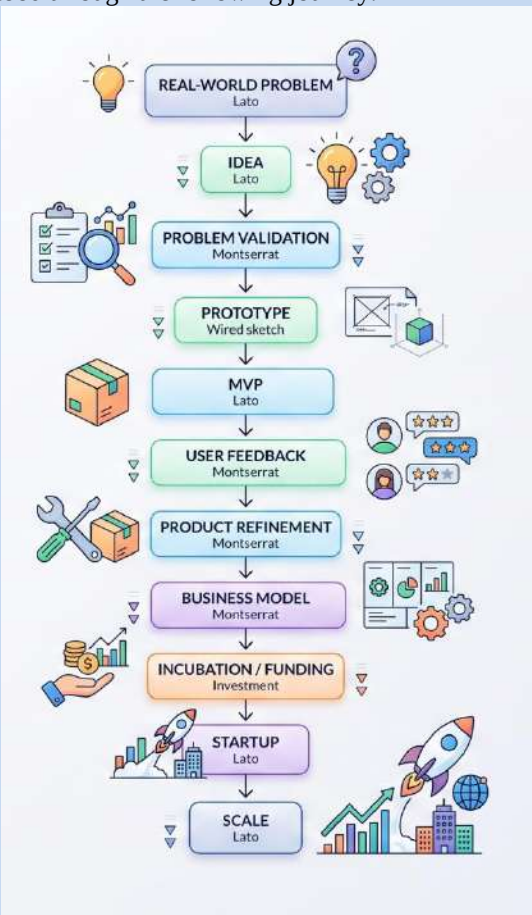


Figure 1: College Project to Startup Journey

The important point is that **technology is only one component**. A startup succeeds when technology, user needs and business viability come together.

Project vs. Product vs. Startup

Aspect	College Project	Product	Startup
Primary objective	Academic learning	Solve a user problem	Create sustainable value
Main focus	Technical implementation	User experience	Customers + technology + business
Users	Faculty/evaluators	Target users	Paying/active customers
Success measure	Project evaluation	User satisfaction	Growth and sustainability
Development	Fixed academic timeline	Iterative	Continuous
Revenue	Usually none	May or may not exist	Usually required for sustainability

A college project demonstrates **technical capability**, while a startup must demonstrate **real-world value**.

1. Start with a Real Problem

Successful technology startups generally begin with a genuine problem rather than a fashionable technology.
Instead of saying:
"We want to develop an application using Generative AI."
students should ask:
"What problem can Generative AI solve better, faster or more affordably?"
Potential problem areas include healthcare, education, agriculture, cybersecurity, transportation, financial inclusion, waste management and accessibility.

Problem Identification Framework

Observe
↓
Identify Pain Point
↓
Talk to Users
↓
Understand Root Cause
↓
Define Problem Statement
↓
Identify Possible Solution

Students should avoid assuming that their first idea is automatically the right problem.

2. Validate the Idea Before Building

One of the most common mistakes made by student entrepreneurs is developing a complete system before confirming whether people actually need it.
Problem validation can involve:

- User interviews
- Surveys
- Observation
- Competitor analysis
- Prototype demonstrations
- Landing pages
- Expert feedback

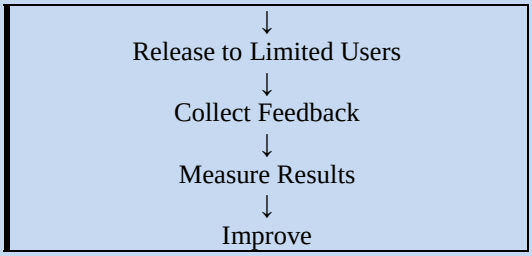
For example, students developing an inventory-management application for small retailers may discover through interviews that their primary problem is actually **cash-flow management**.
Such feedback can save months of unnecessary development.

3. Develop a Minimum Viable Product

After validating the problem, students should develop a **Minimum Viable Product (MVP)**—the simplest functional version capable of testing the core assumption.
The Lean Startup methodology emphasizes experimentation, validated learning and iterative development [2].

MVP Development Cycle

Idea
↓
Smallest Useful Feature Set
↓
Build MVP



An MVP should not attempt to contain every possible feature. The goal is **learning from real users as quickly as possible**.

4. Build a Complementary Team

A startup requires more than programming expertise. A strong student team should ideally combine technical, creative, business and domain skills.

Role	Typical Responsibility
Technical Lead	Architecture, development and technology
AI/Data Specialist	ML models, analytics and data processing
UI/UX Lead	User interface and user experience
Business Lead	Business model, market and customers
Marketing Lead	Branding, promotion and customer acquisition
Domain Expert	Understanding industry-specific requirements

One student may perform multiple roles initially, but the team should collectively cover these capabilities.

5. Use College Resources

Educational institutions can provide valuable resources for student innovators:

- Faculty mentorship
- Laboratories
- Innovation and incubation centres
- Entrepreneurship cells
- Hackathons
- Industry interaction
- Alumni networks
- Intellectual property guidance
- Cloud and software resources
- Startup competitions

The **Atal Innovation Mission (AIM)** has established programs such as Atal Incubation Centres to support innovation and entrepreneurship [3].

Students should actively approach their faculty mentors and institutional incubation or innovation cells.

Role of a Faculty Mentor

Faculty members can therefore become important bridges between **academic learning and entrepreneurship**.

6. Protect Intellectual Property

Before publicly releasing an innovative technology, students should consider intellectual property (IP) protection. Depending on the innovation, protection may involve:

IP Type	Possible Application
Patent	Novel technical invention
Copyright	Software code, documentation, creative content
Trademark	Brand or product name
Trade Secret	Confidential algorithms, processes or business information
Design Protection	Certain aspects of product design

The **Indian Patent Office** provides resources and procedures for protecting inventions in India [4].

Students should also check their institution's intellectual property policy before commercializing a project developed using institutional resources.

7. Build, Measure and Learn

Once an MVP is available, real users should interact with it.

Useful measurements include:

- Number of active users
- User retention
- Customer satisfaction
- Task completion rate
- Conversion rate
- Product usage frequency
- Customer acquisition cost
- Revenue, where applicable

A large number of downloads does not necessarily indicate success. If 1,000 people download an application but only 20 continue using it, the product requires improvement.

The startup mindset therefore follows:

Build → Measure → Learn → Improve

This iterative philosophy is central to the Lean Startup approach [2].

8. Develop a Sustainable Business Model

A useful technology solution must eventually have a sustainable operating model.

Business Model	Example
Subscription	Monthly/yearly access
Freemium	Basic free + premium features
SaaS	Cloud-based software service
B2B	Selling to organizations
B2C	Selling directly to consumers
Licensing	Licensing technology to others
Transaction Fee	Commission on transactions

The **Business Model Canvas** developed by Osterwalder and Pigneur provides a useful framework for examining customers, value propositions, channels, revenue streams, resources and other business components [6].

9. Incubation and Funding

After demonstrating technical and market potential, student founders can explore:

College Project
↓
Innovation Cell
↓
Incubation Centre
↓
Prototype / MVP
↓
Startup Competitions
↓
Government Grants
↓

Angel / Seed Funding
↓
Market Expansion

Possible funding sources include government schemes, institutional grants, innovation competitions, angel investors, venture capital and accelerators.
However, **funding should not be considered the first objective**. A validated problem, useful product and evidence of user interest are more valuable starting points.

10. Scale Responsibly
A prototype that works for 20 users may not work for 20,000 users. Scaling requires attention to:

- Cloud infrastructure
- Database performance
- Cybersecurity
- Data privacy
- System reliability
- Customer support
- Legal compliance
- Financial management

Security and privacy should be considered from the beginning, particularly when applications process health, financial, educational or personal information.

A Hypothetical Example: From Project to EdTech Startup
Consider a team developing an **AI-based placement preparation platform**.

Stage	Student Activity
Problem	Students struggle with placement preparation
Initial Idea	AI-based resume analysis
Validation	Interview students and placement officers
Discovery	Students also need skill-gap analysis
MVP	Resume + skill-gap analysis + mock interview
Pilot	Test with 100 students
Feedback	Students request personalized preparation
Product	Add personalized learning pathways
Business Model	Freemium + institutional subscription
Incubation	Approach college incubation centre
Scaling	Expand to other institutions

The original project has now evolved into a potential **EdTech product and startup**.
Common Mistakes to Avoid

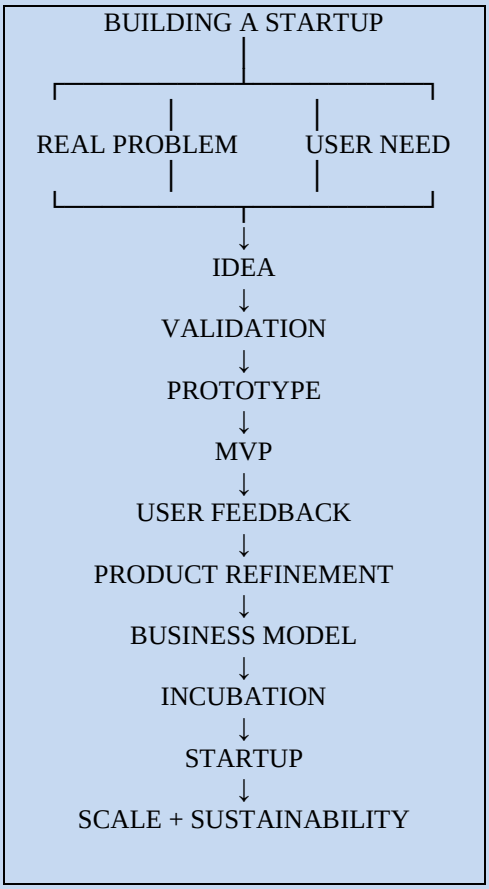
Mistake	Better Approach
Building without validation	Talk to users first
Focusing only on technology	Focus on the problem
Adding too many features	Start with an MVP
Ignoring competition	Study existing solutions
Seeking funding too early	Demonstrate user value
Ignoring security	Apply secure-by-design principles
Working without a team structure	Assign complementary roles
Giving up after failure	Treat failure as feedback

The Role of IT Students in India's Startup Ecosystem
India's rapidly expanding digital economy creates significant opportunities for technology entrepreneurs. AI, cloud computing, IoT, cybersecurity, blockchain, robotics and data analytics provide powerful tools for solving local and global problems.
Students should therefore view academic projects as opportunities to **experiment, innovate and create**, rather than merely as assignments to be completed.
A project that addresses a genuine problem can potentially become:
Project → Research → Prototype → Product → Startup → Social/Economic Impact
Not every project needs to become a company. Even when commercialization does not occur, students gain valuable experience in innovation, teamwork, communication, problem solving and technology development.

Conclusion

A college project should not necessarily end with a project report and viva. It can become the starting point for a product, a research contribution, an innovation or even a technology startup.

The complete journey can be summarized as:



For IT students, entrepreneurship is not necessarily about immediately creating a billion-dollar company. It is about developing the ability to **identify problems, create solutions, test assumptions and generate value**.

The next successful technology startup may not emerge from a large corporate laboratory. It may begin in a college computer lab—with a few students, a faculty mentor, a laptop and a simple question:

"Can we build something that genuinely makes a difference?"

That question can transform an ordinary academic project into an extraordinary journey of **innovation, entrepreneurship and impact**.

References

[1] Department for Promotion of Industry and Internal Trade (DPIIT), Government of India, **Startup India**, Ministry of Commerce and Industry, Government of India.

[2] E. Ries, *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*, Crown Business, 2011.

[3] Atal Innovation Mission, NITI Aayog, Government of India, **Atal Incubation Centers**.

[4] Office of the Controller General of Patents, Designs & Trade Marks, Government of India, **Intellectual Property India**.

[5] S. Blank and B. Dorf, *The Startup Owner's Manual: The Step-by-Step Guide for Building a Great Company*, K&S Ranch, 2012.

[6] A. Osterwalder and Y. Pigneur, *Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers*, John Wiley & Sons, 2010.

[7] World Intellectual Property Organization (WIPO), **Intellectual Property for Start-ups**, WIPO.

[8] Ministry of Education, Government of India, **National Innovation and Startup Policy 2019 for Students and Faculty**, Government of India.

Neuromorphic Computing: Building Brain-Inspired Computers for the Future

Mr. Abhishek Tripathi
Assistant Professor, Department of Information Technology

Introduction

For more than half a century, computing has largely followed the **von Neumann architecture**, in which processing and memory are physically separated. Although this architecture has powered extraordinary advances in computing, modern applications such as Artificial Intelligence (AI), robotics, autonomous vehicles and Internet of Things (IoT) systems increasingly demand enormous computational power with very low energy consumption.

The human brain offers a fundamentally different model. It processes information through billions of interconnected neurons and synapses while consuming remarkably little power compared with conventional high-performance computing systems. This has inspired a new computing paradigm known as **Neuromorphic Computing**.

Neuromorphic computing attempts to design hardware and software inspired by the structure and operating principles of biological neural systems. Rather than continuously moving data between separate memory and processing units, neuromorphic systems aim to perform computation close to where information is stored and process events efficiently [1].

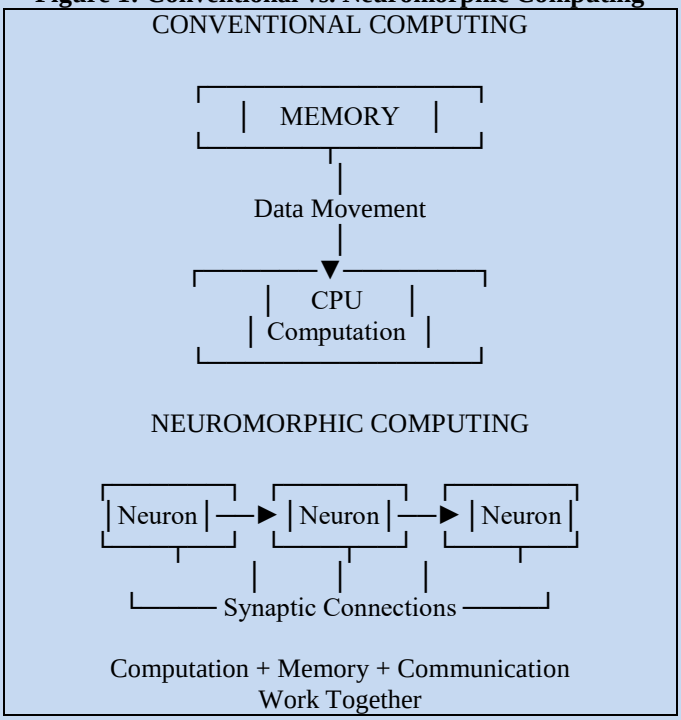
The field is particularly important for the future of AI because increasingly capable AI models require enormous computing resources. Neuromorphic computing offers the possibility of building intelligent systems that can operate with **high efficiency, low latency and extremely low power consumption**.

What is Neuromorphic Computing?

The term *neuromorphic* was introduced by Carver Mead in the late 1980s to describe electronic systems inspired by biological neural structures [2].

A conventional computer generally performs operations using a processor, memory and communication buses. Neuromorphic systems instead attempt to reproduce characteristics of biological neural networks using electronic circuits, specialized processors and learning algorithms.

Figure 1: Conventional vs. Neuromorphic Computing



This architecture reduces unnecessary data movement and enables highly parallel computation.

Key Characteristics

Neuromorphic computing is based on several important principles.

Characteristic	Description
Brain-inspired	Architecture is inspired by neurons and synapses
Event-driven	Computation occurs primarily when events occur
Parallel processing	Many processing elements operate simultaneously
Low power	Designed for energy-efficient computation
Local memory	Memory and computation are closely integrated
Adaptive	Systems can learn or adapt from incoming information
Low latency	Useful for real-time applications

One of the most important concepts is **event-driven computation**. Traditional processors frequently execute instructions according to a clock, even when little meaningful information is changing. Neuromorphic systems can remain largely inactive until meaningful input events occur.

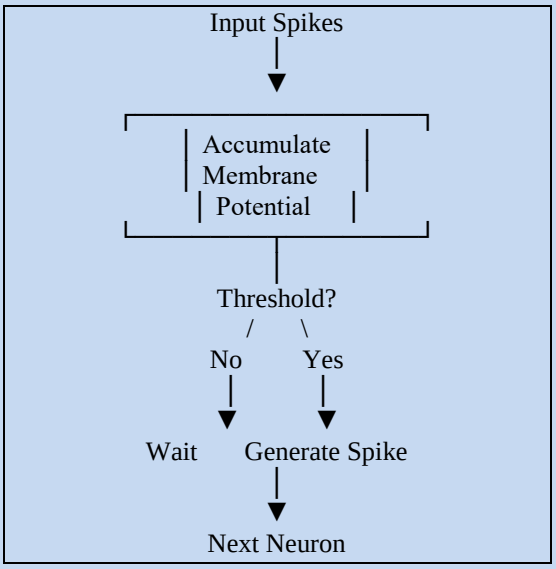
This makes them particularly attractive for battery-powered and continuously sensing systems.

Spiking Neural Networks: The Computational Model

A major component of neuromorphic computing is the **Spiking Neural Network (SNN)**.

Traditional artificial neural networks generally represent information using continuous numerical values. SNNs represent information through discrete electrical-like events called **spikes**.

A simplified neuron can be represented as:



When the accumulated membrane potential reaches a threshold, the neuron generates a spike that can be transmitted to other neurons.

This event-driven model can significantly reduce unnecessary computation, particularly for sparse sensory data.

Neuromorphic Hardware

Researchers and technology organizations have developed specialized neuromorphic processors.

Platform	Organization	Major Feature
Loihi / Loihi 2	Intel	Research-oriented neuromorphic processor
TrueNorth	IBM	Large-scale neurosynaptic architecture
SpiNNaker	University of Manchester	Large-scale spiking neural simulation
BrainScaleS	European research consortium	Accelerated neuromorphic systems
Akida	BrainChip	Edge-oriented neuromorphic AI

Intel's Loihi research platform, for example, was designed around asynchronous spiking neural networks and demonstrated the potential of neuromorphic systems for highly efficient AI workloads [3].

IBM's TrueNorth demonstrated an architecture containing approximately one million programmable neurons and hundreds of millions of synapses, highlighting the scalability of neuromorphic approaches [4].

Neuromorphic Computing vs. Traditional AI Hardware

Feature	CPU/GPU-Based AI	Neuromorphic Computing
Basic computation	Numerical operations	Event/spike-based operations
Architecture	Mostly clock-driven	Often asynchronous/event-driven
Memory	Often separate from computation	Closely integrated
Parallelism	High, especially GPUs	Highly distributed
Energy efficiency	Can be high but workload-dependent	Potentially extremely high for sparse workloads
Latency	Depends on architecture	Very low for suitable real-time tasks
AI models	CNNs, Transformers, etc.	Primarily SNNs and event-driven models
Best suited for	Large-scale model training/inference	Edge, sensing and real-time AI

Neuromorphic computing should not be considered a direct replacement for CPUs or GPUs. Instead, it represents an alternative architecture that may be particularly effective for specific workloads.

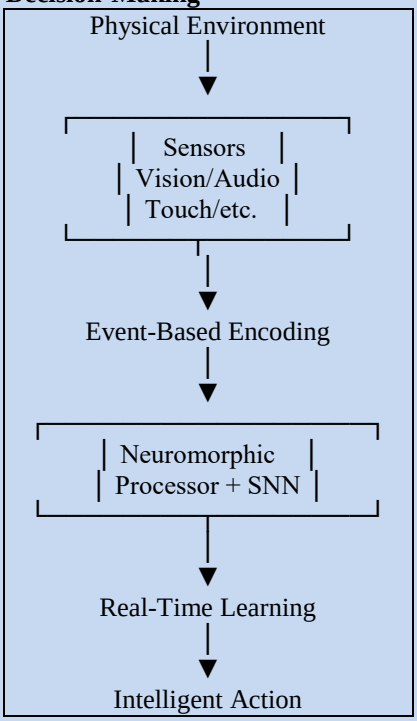
Why Neuromorphic Computing Matters for AI

The rapid growth of AI has created a major challenge: **computational efficiency**.

Large AI models require significant computing resources for training and inference. Data centres consume substantial amounts of electricity, while edge devices such as smart cameras, drones and wearable devices operate under strict power constraints.

Neuromorphic systems address this challenge by attempting to reduce unnecessary computation and data movement.

Figure 2: From Sensing to Intelligent Decision-Making



This architecture is especially attractive when systems must sense and react continuously with limited energy.

Applications of Neuromorphic Computing

1. Robotics

Robots need to process sensory information rapidly while operating with limited energy. Neuromorphic processors can support event-based vision, motion detection, navigation and adaptive control.

2. Autonomous Vehicles

Autonomous systems continuously process camera, radar and other sensor information. Event-based neuromorphic vision can potentially provide low-latency perception with reduced power consumption.

3. Internet of Things

IoT devices often operate on batteries. Neuromorphic computing can allow local intelligence without continuously transmitting data to cloud servers.

4. Healthcare and Wearable Devices

Wearable sensors generate continuous streams of physiological information. Neuromorphic systems could help detect abnormal patterns locally while reducing communication and energy requirements.

5. Smart Surveillance

Event-driven vision systems can identify movement or unusual activity without continuously processing every video frame, potentially reducing computational requirements.

6. Edge AI

Neuromorphic computing is particularly promising for **Edge AI**, where data is processed close to its source rather than being continuously transferred to centralized cloud infrastructure.

Advantages and Challenges

Advantages	Challenges
Very low energy potential	Immature ecosystem
Low-latency processing	Difficult programming models
Massive parallelism	Limited standardization
Suitable for edge devices	SNN training remains challenging
Efficient event processing	Hardware availability
Potential for adaptive systems	Integration with existing AI frameworks

One major challenge is that the conventional software ecosystem is primarily designed for CPUs and GPUs. Neuromorphic systems require specialized programming models, algorithms and development tools.

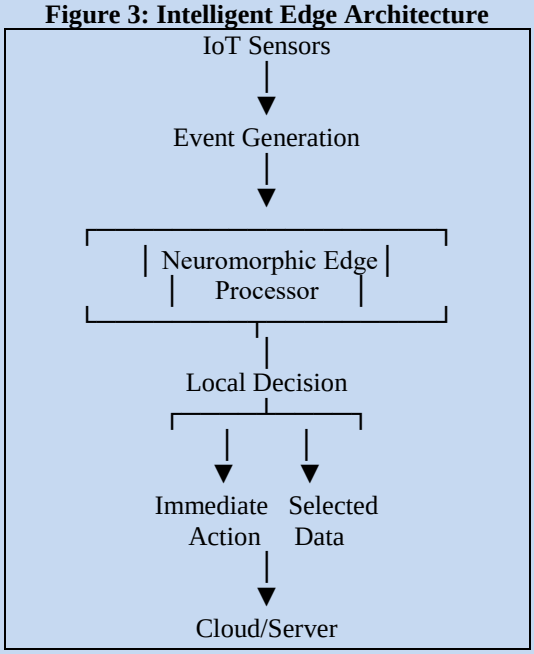
Another challenge is training SNNs effectively. Techniques such as surrogate-gradient learning, temporal coding and conversion of conventional neural networks to SNNs are active research areas.

Neuromorphic Computing and Edge Intelligence

One of the most promising combinations is:

Neuromorphic Computing + Edge AI + IoT

Figure 3: Intelligent Edge Architecture



Instead of transmitting all raw data to the cloud, an intelligent edge device can process information locally and transmit only important results.

This can reduce **latency, bandwidth requirements and energy consumption**, while potentially improving privacy because sensitive data can remain closer to its source.

What Should IT Students Learn?

Neuromorphic computing is interdisciplinary. Students interested in this field should build knowledge in:

Area	Important Topics
Programming	Python, C/C++, embedded programming
AI/ML	Neural networks, deep learning
SNN	Spike coding, neuron models, learning rules
Mathematics	Linear algebra, probability, optimization
Electronics	Digital circuits, VLSI, sensors
Hardware	Embedded systems, accelerators

Edge Computing	IoT, distributed processing
Research	Experimentation and benchmarking

Students can begin with conventional neural networks and gradually explore SNN frameworks and neuromorphic platforms.

The Future of Neuromorphic Computing

Neuromorphic computing is unlikely to replace conventional computing completely. Instead, the future is likely to involve **heterogeneous computing**, where CPUs, GPUs, AI accelerators and neuromorphic processors work together.

For example:

CPU → General Computing

GPU → Large AI Workloads

NPU → AI Inference

Neuromorphic Processor → Event-Based Edge Intelligence

Quantum Processor → Specialized Quantum Workloads

Such systems could dynamically assign workloads to the most appropriate architecture.

The convergence of neuromorphic computing with **AI, robotics, IoT, edge computing and advanced sensors** could enable machines that perceive and respond to their environments with greater efficiency.

Conclusion

Neuromorphic computing represents a fundamental rethinking of how computers process information. Inspired by the brain's neurons and synapses, it emphasizes **event-driven computation, massive parallelism, local memory and energy efficiency**.

Although the technology is still developing and faces challenges in hardware, software, algorithms and standardization, its potential is significant. Applications ranging from autonomous robots and smart sensors to healthcare wearables and intelligent IoT devices could benefit from this approach.

For IT students, neuromorphic computing offers an exciting research direction at the intersection of **Artificial Intelligence, Computer Architecture, Embedded Systems and Neuroscience**.

The future of computing may not simply be about making processors faster. It may be about making them **smarter, more adaptive and dramatically more energy efficient**.

The central idea behind neuromorphic computing can therefore be expressed simply:

Instead of making computers work harder, let us make them compute more like intelligent biological systems—only when computation is needed, where it is needed, and with as little energy as possible.

References

[1] G. Indiveri and S.-C. Liu, "Memory and Information Processing in Neuromorphic Systems," *Proceedings of the IEEE*, vol. 103, no. 8, pp. 1379–1397, 2015.

[2] C. A. Mead, "Neuromorphic Electronic Systems," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1629–1636, 1990.

[3] M. Davies et al., "Loihi: A Neuromorphic Manycore Processor with On-Chip Learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, 2018.

[4] F. Akopyan et al., "TrueNorth: Design and Tool Flow of a 65 mW 1 Million Neuron Programmable Neurosynaptic Chip," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 10, pp. 1537–1557, 2015.

[5] S.-C. Liu and T. Delbruck, "Neuromorphic Sensory Systems," *Current Opinion in Neurobiology*, vol. 20, no. 3, pp. 288–295, 2010.

[6] W. Maass, "Networks of Spiking Neurons: The Third Generation of Neural Network Models," *Neural Networks*, vol. 10, no. 9, pp. 1659–1671, 1997.

[7] E. Neftci, H. Mostafa and F. Zenke, "Surrogate Gradient Learning in Spiking Neural Networks," *IEEE Signal Processing Magazine*, vol. 36, no. 6, pp. 61–63, 2019.

[8] M. Davies, A. Wild, G. Orchard, Y. Sandamirskaya, G. A. Fonseca Guerra, P. Joshi, P. Plank and S. Risbud, "Advancing Neuromorphic Computing with Loihi: A Survey of Results and Outlook," *Proceedings of the IEEE*, vol. 109, no. 5, pp. 911–934, 2021.

Modern Software Testing Using AI: From Test Automation to Intelligent Quality Engineering

Ms. Priya Sharma
Assistant Professor, Department of Information Technology

Introduction

Software has become an essential part of almost every aspect of modern life. From banking and healthcare to education, transportation and entertainment, software systems are expected to be reliable, secure and available around the clock. As applications become more complex and development cycles become shorter, traditional software testing approaches are facing increasing challenges.

Artificial Intelligence (AI), particularly **Machine Learning (ML) and Generative AI**, is transforming software testing by helping testers and developers generate test cases, identify defects, predict risky areas, analyze failures and automate repetitive activities.

The emergence of Large Language Models (LLMs) has further expanded these possibilities. AI can now understand source code, generate test scripts, explain failures and even suggest fixes. NIST has recently initiated dedicated evaluations of AI-generated software tests, including a Code Challenge designed to measure how effectively AI systems can verify the correctness of Python code and generate useful tests [1].

However, AI does not eliminate the need for software testers. Instead, it is changing their role—from writing every test manually to **designing intelligent testing strategies, validating AI-generated results and focusing on complex quality risks**.

Traditional Testing vs. AI-Powered Testing

Traditional testing relies heavily on predefined test cases, rules and human-designed scenarios. AI-powered testing introduces systems capable of learning from code, test history, user behaviour and previous defects.

Aspect	Traditional Testing	AI-Powered Testing
Test creation	Mostly manual/rule-based	AI-assisted generation
Regression testing	Repeated execution of predefined tests	Intelligent test selection/prioritization
Defect prediction	Based mainly on historical analysis	ML-based risk prediction
Test maintenance	Manual updates	Potentially self-healing
Test data	Manually designed	AI-generated/synthetic data
Failure analysis	Human investigation	AI-assisted diagnosis
Code understanding	Human interpretation	LLM-assisted analysis
Human role	Test execution and analysis	Strategy, validation and decision-making

The objective is not simply to replace conventional testing but to **augment human capabilities with intelligent automation**.

How AI Is Transforming Software Testing

1. AI-Based Test Case Generation

Creating comprehensive test cases can be time-consuming, particularly for large applications.

Generative AI tools can analyze source code, requirements and API specifications and suggest test cases automatically.

For example, given a Python function:

calculate_discount(price, customer_type)

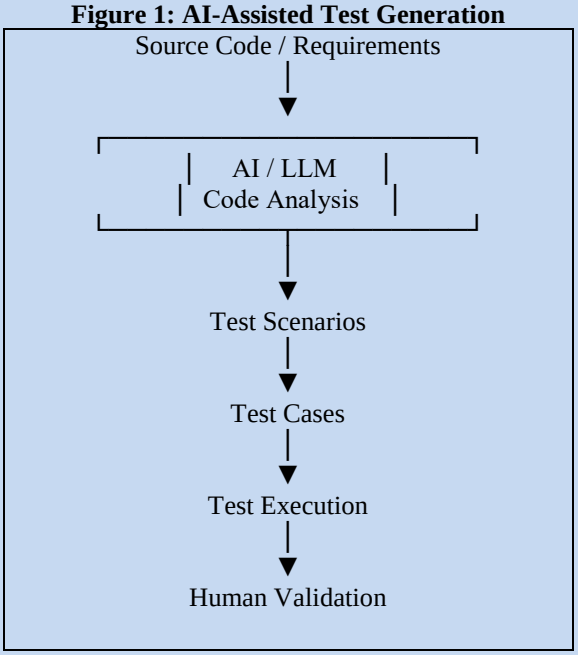
an AI system may generate scenarios for:

- Regular customer
- Premium customer
- Zero price
- Negative price
- Extremely large value
- Invalid customer type
- Null input

This allows developers to discover scenarios that might otherwise be overlooked.

NIST's 2025–26 GenAI Code evaluation specifically examines whether AI systems can generate effective tests for software code, demonstrating the growing importance of AI-based test generation [1].

Figure 1: AI-Assisted Test Generation



---

2. Intelligent Test Data Generation

Testing software with only a few manually selected inputs may leave important conditions uncovered.

AI can generate synthetic test data representing different scenarios, including:

- Boundary values
- Invalid inputs
- Rare combinations
- Large datasets
- Randomized inputs
- User behaviour patterns

This is especially useful for applications where realistic datasets are difficult to obtain because of privacy restrictions. However, AI-generated data must itself be validated. Poor-quality or biased synthetic data can lead to misleading test results.

---

3. Intelligent Regression Testing

Regression testing verifies that changes to an application have not introduced new defects.

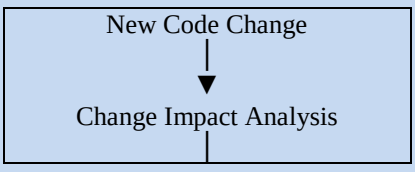
Large projects may contain thousands of automated tests. Running all of them after every small code change can be expensive.

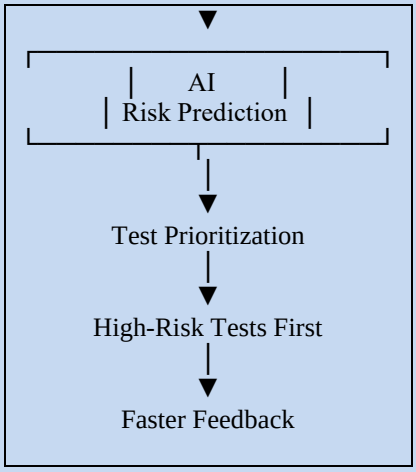
AI can analyze:

- Changed files
- Code dependencies
- Historical failures
- Defect-prone modules
- Test execution history

and prioritize the tests most likely to detect defects.

Figure 2: AI-Based Regression Testing





Research on AI-powered testing indicates that intelligent test generation and prioritization can improve testing efficiency while increasing coverage [2].

---

4. Self-Healing Test Automation

Automated UI tests frequently fail because application interfaces change.

For example, a test may search for:

<button id="login">

but a new application version may replace the identifier with:

<button id="user-login">

A traditional automation framework may report a failure even though the application is functioning correctly.

AI-powered testing tools can analyze the changed interface and attempt to identify the corresponding element automatically.

This concept is known as **self-healing test automation**.

Traditional Automation	Self-Healing Automation
Locator changes cause failure	AI searches for alternative element
Manual script correction	Automated adaptation may occur
High maintenance effort	Reduced maintenance
Fixed rules	Context-aware analysis

Self-healing does not mean that every failure should be automatically repaired. Critical changes should still be reviewed by a human tester.

---

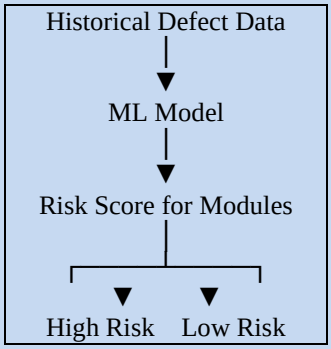
5. AI-Assisted Bug Detection and Prediction

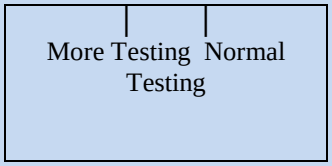
AI can learn from historical defect information to identify areas of software that are more likely to contain defects.

Potential input features include:

- Code complexity
- Number of previous defects
- Frequency of code changes
- Number of developers modifying a module
- Test coverage
- Dependency relationships

A simplified process is:



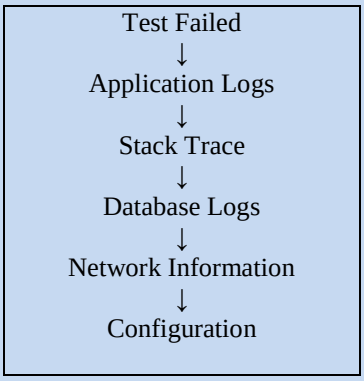


This approach allows testing teams to focus resources where they are most needed.

---

6. AI for Failure Analysis

A failed test does not always identify the actual cause of a problem.
For example:

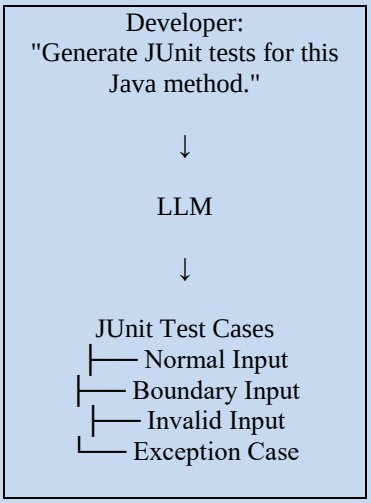


Analyzing all these sources manually can be difficult.
AI systems can summarize logs, identify patterns, correlate errors and suggest possible root causes.
Recent research into LLM-based automated program repair also highlights the potential for AI to support developers in identifying and fixing software defects, while emphasizing the need for human-AI collaboration [3].

---

7. Generating Unit Tests with LLMs

Large Language Models can generate unit tests from functions, classes or modules.
For example:



AI-generated tests can accelerate development, but generated tests should not automatically be assumed to be correct. A test may execute successfully while failing to detect an actual defect. Therefore, developers must evaluate whether the generated tests contain meaningful **assertions, edge cases and expected outcomes**.
Research examining AI assistants for test development suggests that AI-generated tests can be useful, while emphasizing the continuing importance of developer review [4].

---

8. Testing AI-Based Applications

An important development is that AI is not only being used **for testing software**; AI systems themselves must also be tested.

Traditional software generally produces deterministic outputs for given inputs. AI systems may produce probabilistic or context-dependent outputs.
Therefore, AI applications require additional testing dimensions.

Testing Dimension	Example
Accuracy	Is the prediction correct?
Robustness	Does the model handle unusual inputs?
Bias	Does performance vary across groups?
Fairness	Are decisions equitable?
Security	Can the model be manipulated?
Privacy	Does it expose sensitive information?
Explainability	Can decisions be understood?
Reliability	Does performance remain stable?
Safety	Can incorrect outputs cause harm?

NIST's AI Test, Evaluation, Validation and Verification (TEVV) work emphasizes systematic measurement and evaluation of AI systems, while its ARIA program has explored model testing, red teaming and field testing [5][6].

---

9. Testing Generative AI Applications

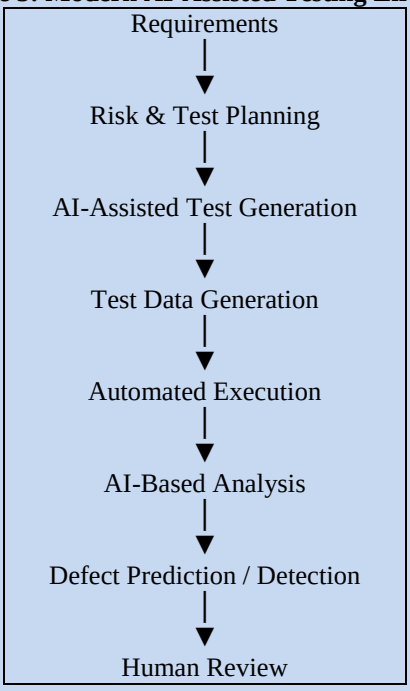
Generative AI applications create an additional challenge because the same prompt may produce different responses. Testing therefore needs to evaluate more than simple pass/fail conditions.
For an AI chatbot, testers may examine:

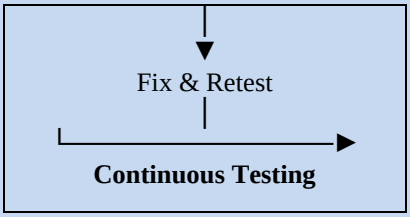
- Factual accuracy
- Relevance
- Consistency
- Toxicity
- Hallucinations
- Prompt injection resistance
- Privacy leakage
- Instruction following
- Response safety

NIST's GenAI evaluation program uses generators, discriminators and prompting strategies to investigate the capabilities and limitations of generative AI systems across modalities including text, image and code [7].
This represents an important shift from traditional software testing toward **AI evaluation and continuous monitoring**.

AI Testing Lifecycle

Figure 3: Modern AI-Assisted Testing Lifecycle





Modern testing therefore becomes a continuous activity integrated into the Software Development Life Cycle (SDLC) and DevOps pipeline.

Benefits of AI-Powered Software Testing

Benefit	Impact
Faster test creation	Reduces repetitive manual work
Higher test coverage	Identifies additional scenarios
Intelligent prioritization	Focuses testing on risky areas
Faster debugging	Helps analyze logs and failures
Reduced maintenance	Supports adaptive automation
Continuous testing	Integrates testing into CI/CD
Better scalability	Handles large test suites
Productivity improvement	Allows testers to focus on complex tasks

AI can therefore shift testing from a predominantly **execution-oriented activity to an intelligence-oriented engineering discipline**.

Challenges and Limitations

AI-powered testing is not without risks.

1. Incorrect AI Output

An LLM can generate tests that appear reasonable but contain incorrect assumptions.

2. False Confidence

A large number of AI-generated tests does not necessarily mean good coverage.

3. Security and Privacy

Source code, logs or test data sent to external AI services may contain sensitive information.

4. Bias

AI-based testing tools may inherit biases from their training data.

5. Explainability

Testers may not always understand why an AI system selected a particular test.

6. Human Oversight

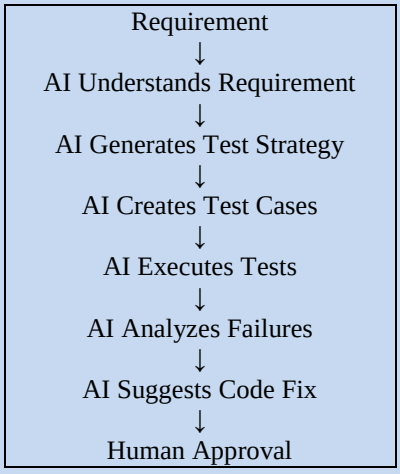
AI should support—not replace—professional judgment.

NIST's research on AI testing also emphasizes that evaluating AI systems requires appropriate measurements because conventional software testing metrics may not fully capture AI behaviour [8].

The Future of AI in Software Testing

The future is likely to involve **AI-assisted autonomous testing**, in which intelligent agents can analyze requirements, generate tests, execute them, interpret results and recommend fixes.

A possible future workflow is:



↓
Automated Retesting
↓
Deployment

However, completely autonomous testing should be approached carefully, especially for safety-critical systems such as healthcare, aviation, transportation and financial infrastructure.
The most realistic future is therefore **human + AI collaboration**, where AI handles scale and repetitive analysis while humans provide context, judgment and accountability.

What IT Students Should Learn
For students aspiring to careers in software testing and quality engineering, the skill set is evolving.

Traditional Skills	Emerging AI Testing Skills
Manual testing	AI-assisted testing
Selenium/automation	LLM-assisted test generation
SQL	Test-data generation
API testing	AI API evaluation
Unit testing	AI-generated unit tests
Bug tracking	AI-based defect analysis
Test design	Prompt engineering
CI/CD	AI-integrated DevOps
Functional testing	AI safety and robustness testing

Students should therefore learn both **software testing fundamentals and AI technologies**. Understanding test design, equivalence partitioning, boundary-value analysis, regression testing and automation remains essential even when AI tools are used.

Conclusion
Artificial Intelligence is transforming software testing from a largely rule-driven activity into a more **adaptive, predictive and intelligent discipline**.
AI can generate test cases, create test data, prioritize regression tests, detect defects, analyze failures and assist with program repair. Generative AI is further enabling testers to interact with software systems using natural language and automatically produce test artifacts.
At the same time, AI introduces new testing challenges. AI-generated tests may be incomplete or incorrect, while AI-based applications themselves require testing for accuracy, robustness, fairness, security, privacy and safety.
The future therefore does not belong to **AI replacing software testers**. It belongs to testers who know how to work effectively with AI.
For IT students, this creates an exciting opportunity. The software tester of the future will not simply ask:
"Does the software work?"
Instead, the question will become:
"How intelligently, securely, reliably and continuously can we demonstrate that the software works?"
The answer will increasingly come from the collaboration of **human expertise, automated testing and Artificial Intelligence**.

References
[1] P. Fontana, Y. Lee, H. Iyer and S. Sharma, *2025 NIST GenAI (Pilot): Code Challenge Evaluation Plan*, National Institute of Standards and Technology, 2025/2026.
[2] M. Baqar and R. Khanda, "The Future of Software Testing: AI-Powered Test Case Generation and Validation," *arXiv preprint*, 2024.
[3] L. Zhang, A. Singhal, Q. Zou, X. Sun and P. Liu, "Can AI Fix Buggy Code? Exploring the Use of Large Language Models in Automated Program Repair," *IEEE Computer*, 2025.
[4] V. Joshi and I. Band, "Disrupting Test Development with AI Assistants," *arXiv preprint*, 2024.
[5] National Institute of Standards and Technology (NIST), "AI Test, Evaluation, Validation and Verification (TEVV)," NIST, 2026.
[6] R. Amironesei et al., *Assessing Risks and Impacts of AI (ARIA): Pilot Evaluation Report*, NIST AI 700-2, 2025.
[7] National Institute of Standards and Technology (NIST), "NIST Generative AI Evaluation Program," 2026.
[8] E. Lanus et al., "Data Frequency Coverage Impact on AI Performance," *2025 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, pp. 258–267, 2025.
[9] National Institute of Standards and Technology (NIST), "Recommended Minimum Standards for Vendor or Developer Verification (Testing) of Software," NIST, updated 2025.

Tackling Plastic Waste: A Complete Guide to Sustainable Management

By Abhishek Tiwari
B.Tech, Information Technology, 4th Year
Babu Banarasi Das Northern India Institute of Technology

Plastic is everywhere, from packaging that keeps our food fresh to bottles that hold our drinks. However, with convenience comes responsibility. The amount of plastic waste generated any given day calls for efficient management as a means to protect our environment. This article is going to take you on a journey concerning plastic waste, right from collection to its disposal, introducing innovative methods and practices shaping the future of sustainability.

Collection: The First Step in the Journey

Before plastic can be transformed or repurposed, it needs to be collected efficiently. Two dominant methods rule the roost:

Curbside Collection

- **Single-Stream Recycling:** Imagine a world where recycling is as easy as throwing everything into one bin. That's the beauty of single-stream recycling. It's convenient for consumers but demands sophisticated sorting processes further down the line.
- **Dual-Stream Recycling:** A bit more hands-on is the dual-stream recycling approach. In this methodology, residents separate plastics from other recyclables at the source for a cleaner and more efficient recycling process.

Drop-Off Centres

- **Bottle Banks:** Plastic bottle recycling finds its ultimate destinations in these bottle banks. Conveniently situated in public areas, bottle banks ensure that a contribution toward a cleaner planet is only a few steps away.
- **Plastic-Stream Recycling:** For those who want to go the extra mile, these centers accept more types of plastics to make sure that different kinds of plastic waste are dealt with appropriately.

Sorting: Separating the Trash from the Treasure

Collected plastic wastes go to sorting facilities where painstaking separation is done. Why? So each type of plastic is treated most effectively.

Manual Sorting

- **Visual Inspection:** Skilled workers sort plastics by a visual look, identifying types based on colour, shape, or other visible characteristics.
- **Tactile Identification:** Sometimes, a touch of magic is created with just a touch. Workers can feel the texture and rigidity of plastics to ensure each piece finds its right place.

Automatic Sorting

- **Near-Infrared Spectroscopy (NIR):** Just imagine a high-tech system where plastics are lit up to tell what chemical makeup they are made of. NIR is fast, accurate, and a real game-changer when it comes to waste management.
- **X-Ray Fluorescence (XRF):** Some plastics hide their secrets in their chemical composition, and XRF is the technology used to detect them. It means materials like PVC are sorted out with much precision.

Processing: Give Plastic a Second Life

Sorting plastic waste isn't enough; it follows suit by readying it for processing. Now things get exciting: taking refuse material and turning it into useful resources.

Mechanical Recycling

- **Shredding:** The first step in mechanical recycling is shredding the plastic into smaller, more manageable pieces.
- **Melting:** The bits then melt into a moldable shape that can be reshaped.
- **Extrusion:** Finally, plastic melts are extruded into pellets or new products ready to start their life afresh.

Chemical Recycling

- **Depolymerization:** This is the process where the plastics are broken down into simple building blocks. These can then be reassembled into new plastics.
- **Solvolysis:** It is the use of solvents in dissolving and breaking down plastics into their basic building blocks, which subsequently could be employed either as raw materials in producing new plastic products or other useful material forms.
- **Pyrolysis:** This involves the heating up of plastics without the supply of oxygen to produce smaller molecules that could be used as fuels or feedstocks to re-produce other plastics.

End-Use Applications: From Waste to Wonder

Recycled plastic is not all about keeping trash out of landfills but creating useful and appealing products that meet our daily needs.

Packaging

- **Food Containers:** Recycled plastics are best suited for making food containers and help keep our meals fresh along with maintaining cleanliness in the environment.
- **Bottles and Jars:** One of the most usual applications for recycled plastics, these items underscore the capability to change trash into a useful product.

Construction

- **Insulation:** Recycled plastic forms the base for insulation materials, which make a big deal in saving energy in buildings.
- **Pipes and Fittings:** These are durable and resistant to chemicals, hence very suitable for building purposes.

Automotive

All the way from dashboards to bumpers, the use of recycled plastics helps to make lighter fuel-efficient cars.

Textile

Even in fashion, recycled plastics are making waves. Synthetic fibers from recycled materials are used in clothing, upholstery, and more.

Disposal: The Final Step

Even with all these efforts, not all plastic can be recycled. For the waste that remains, disposal methods become important.

Landfilling

- **Sanitary Landfills:** These sites are carefully managed to minimize environmental impact, containing waste in a controlled manner.
- **Bioreactor Landfills:** These landfills are new and innovative, intensifying decomposition processes to degrade organic material with the use of heightened microbial activity.

Incineration

- **Waste-to-Energy:** Incineration can convert plastic waste into energy with volume reduction of landfills and generating electrical energy simultaneously.
- **Cement Kilns:** Sometimes, plastic wastes are used as an alternate fuel in the manufacture of cement—a very realistic and useful environmental solution.

Policies and Regulations: Steering the Ship

Good practices in waste management do not happen by accident but are instead the end result of stringent policies and rules.

- **Extended Producer Responsibility (EPR):** EPR policies ensure that the responsibility for the lifecycle of a product, from its making to its disposal, rests with the manufacturer itself, hence encouraging more environmentally friendly considerations.
- **Recycling Targets:** Ambitious targets for recycling are laid down by governments, which challenge industries to bring in more innovative technologies and investments for easier and greater recycling.

Conclusion: A Way Forward

The processing of plastic waste from its collection to disposal is a long but important journey. Sorting, treatment, and final disposal—all these stages contribute to the reduction of environmental impacts and are fundamental to sustainability. We can make it happen, with good policy measures, turning plastic waste from problem to resource. The implementation of such strategies is crucial for reduced plastic waste, preservation of resources, and better care for our planet for future generations to enjoy.

Serverless Cloud Using AWS Lambda Function

By Awani Shukla
B.Tech, Information Technology, 4th Year
Babu Banarasi Das Northern India Institute of Technology

Project Overview

In today's fast-paced digital landscape, businesses are increasingly adopting serverless architectures to enhance scalability, reduce operational costs, and streamline application development. This project focuses on building a serverless cloud application using AWS Lambda, a key component of Amazon Web Services (AWS) that allows developers to run code without provisioning or managing servers. The goal is to create a scalable, cost-effective, and efficient cloud application that can respond to events in real-time.

Objectives

1. **Understand Serverless Architecture:** Gain insights into the principles of serverless computing and how AWS Lambda fits into this paradigm.
2. **Develop a Serverless Application:** Create a functional application that leverages AWS Lambda to handle backend processes.
3. **Integrate with Other AWS Services:** Utilize other AWS services such as API Gateway, DynamoDB, and S3 to enhance the functionality of the application.
4. **Implement Event-Driven Design:** Design the application to respond to various events, demonstrating the power of serverless computing.
5. **Monitor and Optimize:** Use AWS CloudWatch to monitor the application's performance and optimize its cost and efficiency.

Project Components

1. AWS Lambda

AWS Lambda is the core of our serverless application. It allows us to run code in response to events without the need to manage servers. We will create multiple Lambda functions to handle different tasks, such as processing data, responding to API requests, and interacting with other AWS services.

2. API Gateway

To expose our Lambda functions as RESTful APIs, we will use AWS API Gateway. This service will allow us to create, publish, maintain, and secure APIs at any scale. It will serve as the entry point for client applications to interact with our serverless backend.

3. DynamoDB

For data storage, we will utilize Amazon DynamoDB, a fully managed NoSQL database service. It provides fast and predictable performance with seamless scalability. Our Lambda functions will interact with DynamoDB to store and retrieve data as needed.

4. S3 (Simple Storage Service)

We will use Amazon S3 for storing static assets, such as images, videos, and other files. S3 provides high durability and availability, making it an ideal choice for storing application assets.

Implementation Steps

Step 1: Set Up AWS Account

Create an AWS account if you do not already have one. Ensure that you have the necessary permissions to create and manage Lambda functions, API Gateway, DynamoDB, and S3.

Step 2: Create Lambda Functions

1. Navigate to the AWS Lambda console.
2. Create a new Lambda function for each task (e.g., data processing, API response).
3. Write the code for each function in the supported languages (Node.js, Python, Java, etc.).
4. Configure the necessary permissions and environment variables.

Step 3: Set Up API Gateway

1. Go to the API Gateway console.
2. Create a new API and define the resources and methods (GET, POST, etc.).
3. Integrate the API methods with the corresponding Lambda functions.
4. Deploy the API to make it accessible.

Step 4: Configure DynamoDB

1. Navigate to the DynamoDB console.
2. Create a new table to store application data.
3. Define the primary key and any necessary indexes.
4. Update the Lambda functions to interact with the DynamoDB table.

Step 5: Set Up S3

1. Go to the S3 console.
2. Create a new bucket to store static assets.
3. Upload any necessary files and configure bucket policies for access.

Conclusion

The **"Serverless Cloud Using AWS Lambda Function"** project demonstrates the power and flexibility of serverless architecture. By leveraging AWS Lambda and other AWS services, we can build scalable, cost-effective applications that respond to events in real-time. This project not only enhances our understanding of serverless computing but also equips us with the skills to develop modern cloud applications that meet the demands of today's digital world.