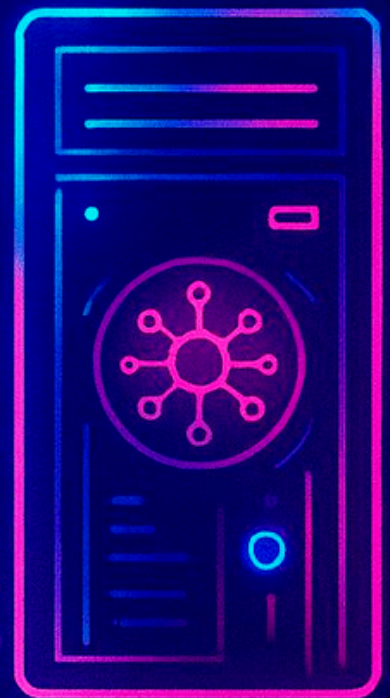


COMPUTECH

10TH EDITION 2024-25



PEEKING INTO THE CODE THAT
POWERS OUR WORLD



GIRIJANANDA
CHOWDHURY
UNIVERSITY

Computech **EDITORIAL BOARD**

EDITOR

MRS. MRIDUSMITA BARUAH

STUDENT EDITOR

**AKASH BORA
DWIJ RAJ BHAGAWATI
AYON CHILA OJA**

COVER PAGE DESIGN

**AYON CHILA OJA
DWIJ RAJ BHAGAWATI**

LAYOUT DESIGN

AKASH BORA



**10TH EDITION, 2024-25 PUBLISHED BY DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING GCU, GUWAHATI-781017**

VISION & MISSION

OF THE UNIVERSITY



VISION

- Girijananda Chowdhury University shall create a culture of education that will provide students a transformative experience - intellectual, ethical, cultural, and social. We envision a society where our students shall contribute with unparalleled professionalism, competence and integrity while upholding the basic human values of liberty, justice, and equality.



MISSION

- Girijananda Chowdhury University embarks upon its academic plan and programs with a mission of empowering our students with knowledge and skills in their chosen domain with a strong mooring in its envisioned values.
- We have a mission to be contemporary in our pedagogy and to be future focused in developing courses and research trajectories considering the dynamic realities of the region, nation and the world at large.
- Our mission is to equip our students with the requisite knowledge, skills and values in a unique ecosystem that will enable them to avail the opportunities coming in the wake of the industry 4.0. In this mission we strive for vibrant industry - academia bonding and excellence in research and innovation.
- We have a mission of creating human resources critical for harnessing the unique and diverse natural resources of the northeast for a strong and vibrant economy necessary for sustainable development of the region.
- Girijananda Chowdhury University's mission of creating citizens who will be globally competent and competitive with skills for communication, collaboration and cooperation is embedded in all its academic programs and research.
- Our mission is to create a cultural continuum with our neighboring countries of Southeast Asia on the foundation of the commonalities of our tradition, culture, and common history.
- Our mission is to enable our students to contribute towards a society that is heterogeneous, cosmopolitan, and egalitarian.

VISION & MISSION

OF DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING



VISION

- To develop competent computer science and engineering graduates who will contribute productively to local, national as well as global economies.



MISSION

- **M1:** To provide good quality education through state-of-the-art facilities and pedagogy in teaching.
- **M2:** To motivate the students to develop research-based, societal projects through innovative solutions and to make them acquainted with the latest developments, traditions, and culture through seminars, workshops, expert talks, and industrial visits.
- **M3:** To promote and nurture a conducive environment in the department so that the students ingrain professional values and ethical behavior to contribute towards a society that is heterogeneous, cosmopolitan, and egalitarian.

PROGRAM OUTCOMES (POS)

Engineering Graduates will be able to:

- **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
- **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.
- **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.
- **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.

PROGRAM OUTCOMES (POS)

- 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.
- 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.
- Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
- Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
- Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PROGRAM OUTCOMES(POS)

- Communication: Communicate effectively on complex engineering activities with the engineering community and with 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.
- 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.
- Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSO_s)

The Computer Science and Engineering graduates are able to

- **PSO1.** Analyze, design, develop, test and apply mathematics, science, engineering fundamentals and management principles to solve complex engineering problems.
- **PSO2.** Develop their skills in the field of computer science and its latest trends for successful career, research, entrepreneurship and higher studies.
- **PSO3.** Apply ethical principles and commit to professional and social responsibilities.

PROGRAM EDUCATIONAL OBJECTIVES (PEO_s)

- **PEO1:** Graduates will be able to apply the mathematical, scientific and engineering fundamentals to solve a wide range of computational problems in industry, government, or other work environments.
- **PEO2:** Graduates will be able to become entrepreneurs, employees of reputed organizations, pursue higher studies including research enhancing their skills in Computer Science and Engineering.
- **PEO3:** Graduates will be able to exhibit professional and ethical attitude, effective communication skills and teamwork for professional development in societal and environmental contexts.

Courses Offered:

- **B.Tech in Computer Science & Engineering**
- **M.Tech in Computer Science & Engineering**
- **Ph.D in Computer Science & Engineering**

| CONTENTS



EDITORIAL

11



ARTICLES

17



ACADEMIC SPOTLIGHT

71



MESSAGE FROM THE DEAN OF ENGINEERING AND TECHNOLOGY

The School of Engineering and Technology plays a crucial role in the education of emerging engineers and individuals with aspirations, fostering a shared ambition to create a positive impact in the world. The faculty of our school has a decade of experience in engineering education and has garnered a significant reputation as a highly sought-after educational institution among students in the surrounding area throughout the course of its existence. Our school has a comprehensive infrastructure, well-equipped laboratories, an extensive library, and a substantial faculty comprised of highly experienced individuals.

As an academic institution, our foremost objective is the continual enhancement of our endeavours. Our primary objective is to inculcate awareness and responsiveness towards the requirements of the community, while concurrently offering valuable contributions to the domains of education, knowledge, and society at large. Our primary objective is to assist students in cultivating the attributes of self-assurance, proficiency, and adaptability that are essential for achieving success in workplace and making meaningful contributions to society.

With best wishes,

Prof. S. Robert Ravi
Professor and Dean,
School of Engineering and Technology,
Girijananda Chowdhury University, Assam

MESSAGE FROM THE HOD OF COMPUTER SCIENCE & ENGINEERING



It gives me immense pleasure to share this edition of “CompuTech”, a proud reflection of the Department of Computer Science & Engineering’s spirit of learning, innovation, and collaboration. The magazine serves as a creative forum for our students and faculty to showcase their technical insights, research endeavors, and artistic expressions.

In this era of rapid technological transformation, our department remains committed to equipping students with strong technical foundations, analytical thinking, and ethical values. We continuously strive to bridge the gap between academia and industry, preparing our graduates to meet global challenges with confidence and competence.

I wholeheartedly appreciate the sincere efforts of the editorial team, faculty mentors, and student contributors who have worked tirelessly to bring this issue to life. May this magazine continue to inspire excellence and creativity in all our future endeavors.

Best wishes,

A handwritten signature in purple ink that reads "Minakshi Gogoi".

Prof. Minakshi Gogoi

Head of Department

Department of Computer Science & Engineering,
Girijananda Chowdhury University, Assam



MESSAGE FROM THE EDITORIAL BOARD

It gives me great pleasure to present the 10 th edition of our technical magazine “CompuTech” from the Department of Computer Science and Engineering, Girijananda Choudhury University, Assam. This magazine embodies the enthusiasm, creativity, and innovative mindset of our students, who have contributed despite the challenges of their academic commitments.

“CompuTech” is not just a platform for technical expression but also a medium through which students showcase their ideas, imagination, and aspirations. It reflects their determination to go beyond classroom learning and contribute meaningfully to the world of knowledge and innovation.

I take this opportunity to express my sincere gratitude to our management, our respected Head of Department Prof.. Minakshi Gogoi, and all faculty and staff for their constant encouragement and guidance. I also appreciate the dedicated efforts of Dwij Raj Bhagawati, Akash Bora and Ayan Oja , student editorial board members, whose support has been invaluable in bringing out this issue.

I hope this magazine inspires our readers to explore, innovate, and dream bigger.

Mrs. Mridusmita Baruah

Laboratory Instructor

Department of Computer Science & Engineering

Girijananda Choudhury University, Assam

MESSAGE FROM THE STUDENT EDITORIAL BOARD



“Programs must be written for people to read, and only incidentally for machines to execute.” – Harold Abelson

It is with great pleasure that I present this edition of our departmental magazine, themed “Peeking into the Code that Powers Our World.” In an age where technology touches every corner of human existence, Computer Science and Engineering stands as the silent architect shaping how we live, learn, and connect. For us as students, coding is more than a skill — it is a way of thinking. Each line we write carries intention, creativity, and the potential to influence lives. Behind every successful innovation lies curiosity, collaboration, and the courage to explore the unknown. This magazine captures that very spirit — showcasing the voices, ideas, and imaginations of our department. From insightful articles to technical explorations, every contribution reflects our shared journey toward innovation and excellence.

“Stay hungry, stay foolish.” – Steve Jobs

As future engineers, may we continue to embrace that hunger — to question, to break boundaries, and to build solutions that make a difference. Let this issue remind us that technology is not just about efficiency, but about empathy — about understanding people as much as machines. I extend my heartfelt gratitude to our Head of the Department, respected Minakshi Gogoi, for the unwavering guidance, encouragement, and inspiration that continually drives our department forward. My sincere thanks also go to our dedicated faculty mentors and the magazine editorial team members, whose hard work, creativity, and teamwork have made this publication possible. **“Talk is cheap.**

Show me the code.” – Linus Torvalds

As we peek into the code that powers our world, may we also discover the purpose that powers our passion.

Ayon Chila Oja

Editor

CSE Department

Girijananda Chowdhury University



MESSAGE FROM THE STUDENT EDITORIAL BOARD

In a world woven together by codes and creativity, Computech emerges as more than a magazine — it is a narrative of innovation, curiosity, and human intellect. Within these pages lies the voice of a generation that dreams in algorithms, builds in logic, and creates with imagination.

As the realm of technology expands beyond the imaginable, Computech becomes our collective canvas — reflecting the evolution of ideas, the courage to question conventions, and the desire to transform vision into reality. Each article, each design, each expression in this edition is a spark of that relentless pursuit of excellence which defines our Department of Computer Science and Engineering.

Being part of Computech has been an experience of weaving together stories that resonate with passion and purpose. It stands as a tribute to every mind that believes technology is not just about machines — it is about meaning.

I extend my heartfelt appreciation to all the contributors, designers, and visionaries who poured their brilliance into this creation. May Computech continue to inspire — to be the bridge between intellect and imagination, between curiosity and creation, between today's learner and tomorrow's innovator.

Dwij Raj Bhagawati

Editor

CSE Department

Girijananda Chowdhury University

MESSAGE FROM THE STUDENT EDITORIAL BOARD



“Technology is best when it brings people together.” – Matt Mullenweg

In an era where code doesn't just run machines but shapes movements, I believe we stand at a powerful intersection - where creativity meets clarity, and innovation meets responsibility. This edition of our departmental magazine is more than a showcase of technical prowess; it's a celebration of the minds that dare to build, question, and refine.

As someone who thrives on transforming complexity into clarity - whether through cross-platform scripting, anime-style visual demos, or beginner-friendly documentation - I've come to see technology not just as a tool, but as a language of empowerment.

You'll find stories of students pushing boundaries, not just in code but in communication, ethics, and design.

Let this magazine be a reminder: the future isn't just written in code. It's written in how we teach, how we share, and how we care. Whether you're debugging your first script or designing your tenth demo, know that your work matters - and that originality, empathy, and ethical intent are your greatest assets.

To our faculty, mentors, and peers - thank you for fostering a space where curiosity is nurtured and collaboration is celebrated. And to every reader: may this issue inspire you to create boldly, document clearly, and always code with purpose.

“The best way to predict the future is to invent it.” – Alan Kay

Akash Bora

Editor

CSE Department

Girijananda Chowdhury University

| ARTICLES

TITLE	AUTHOR	PAGE
INSIDE THE MIND OF A PROGRAMMER	SUVRADEEP DUTTA	19
GAMING & CODING: HOW THEY ARE CONNECTED	KOUSHIK BHATTACHRAYA	21
WHAT IS CODING, REALLY?	ANGSHITA KASHYAP	24
AI: FRIEND OR FOE? AN ADVANCED PERSPECTIVE	DEBASHISH KASHYAP	27
BLOCKCHAIN TECHNOLOGY AND ITS IMPACT ON MODERN BUSINESS	SATAVISHA ROY CHOUDHURY	30
HACKING THE SYSTEM	YUVRAJ SARMA	32
MY SMART INDIA HACKATHON DIARY	AKASH BORA	37
CYBER SECURITY	ANKITA CHAKRABORTY	42

TITLE	AUTHOR	PAGE
THE PROGRESSION OF PROGRAMMING LANGUAGES: FROM C TO CHATGPT	SNEHA MAZUMDER	45
DATA SCIENCE AND DATABASE	ASHUTOSH DUTTA	47
MINI BRAINS AS PROCESSORS: FUTURE OF ORGANIC COMPUTING	AYON CHILA OJA	49
FROM VACUUM TUBES TO QUANTUM CHIPS	BHARGAB TALUKDAR	51
THE INSIDE STORY OF REVOLUTIONARY VERSION CONTROL SYSTEM	ANKUR RANJAN CHANDA	53
FROM ROOM SIZED TO POCKET SUPER COMPUTERS	CHINMOY KALITA	58
FROM ENGINES TO ALGORITHMS: THE EVOLUTION OF MORDERN MACHINES	PRATYUSH B. CHOUDHURY	61
REACT FATIGUE IS REAL. WILL SVELTE TAKE OVER?	RITANJIT DAS	63
ADA LOVELACE - THE WORLD'S FIRST PROGRAMMER	ABHIGYAN BARUAH	66
THE BEAUTY IN BROKEN CODE	DWIJ RAJ BHAGAWATI	68

INSIDE THE MIND OF A PROGRAMMER



SUVRADEEP DUTTA
CSE 2nd SEM

INTRODUCTION

Inside the Mind of a Programmer is a fascinating topic that delves into the thought processes, problem-solving strategies, and creative approaches that programmers use when developing software.

Here are some key aspects to consider:

1. Problem-Solving Approach

- **Analytical Thinking:** Programmers break down complex problems into smaller, manageable parts, analyzing requirements to determine the best approach for each component.
- **Algorithmic Mindset:** They think in terms of algorithms and data structures, ensuring efficiency and scalability when designing solutions.

2. Creativity and Innovation

- **Creative Solutions:** Programming goes beyond writing code—it involves crafting innovative solutions to unique challenges. Programmers often think outside the box to develop efficient and elegant approaches.
- **Experimentation:** Many programmers enjoy testing new technologies, frameworks, and languages, which can lead to creative breakthroughs.

3. Attention to Detail

- **Precision:** A single typo or misplaced character can cause bugs. Programmers cultivate a keen eye for detail, ensuring their code is clean and error-free.

4. Debugging Skills

- **Systematic Approach:** Programmers use various debugging tools and techniques to identify and fix issues in their code efficiently.

5. Continuous Learning

- **Adaptability:** The tech landscape evolves constantly, requiring programmers to stay updated with new languages, tools, and best practices through self-directed learning.
- **Community Engagement:** Many programmers actively participate in online communities, forums, and open-source projects to exchange knowledge and learn from others.

6. Collaboration and Communication

- **Team Dynamics:** Effective communication is essential for programmers working in teams, enabling seamless collaboration with developers, designers, and stakeholders.
- **Code Reviews:** They engage in code reviews, providing and receiving feedback to enhance code quality and share knowledge.

7. Time Management and Productivity

- **Prioritization:** Programmers juggle multiple tasks and projects, requiring strong time-management skills to stay productive.
- **Focus and Flow:** Many programmers experience a "flow" state when deeply immersed in coding, allowing them to produce high-quality work efficiently.

8. User-Centric Thinking

- **Empathy for Users:** Great programmers prioritize the end-user experience, designing intuitive software that meets user needs.
- **Feedback Loops:** They actively seek user feedback to iterate and refine applications, ensuring usability and effectiveness.

CONCLUSION

Understanding the programmer's mindset requires recognizing the blend of analytical skills, creativity, attention to detail, and continuous learning. While each programmer may have their own unique approach, these common themes highlight the depth and complexity of their thought processes.

GAMING & CODING

HOW THEY ARE CONNECTED



KOUSHIK BHATTACHARYA
CSE 2nd SEM

INTRODUCTION

Gaming has become one of the most popular forms of entertainment in the world, captivating people of all ages. But behind every exciting game lies the power of coding. Coding, or computer programming, is what brings a game to life. Whether it's a simple mobile game or a complex virtual reality experience, games depend on code to function. The relationship between gaming and coding is essential, and understanding it opens the door to creativity, learning, and innovation.

The Role of Coding in Game Development

Coding is the foundation of all digital games. Programmers use languages like C++, Python, C#, and JavaScript to create the structure, logic, and functionality of a game. These languages control how characters move, how enemies react, how scores are calculated, and how players interact with the game environment. Game engines such as Unity and Unreal Engine help developers combine different elements like graphics, sound, physics, and artificial intelligence through code. This allows games to run smoothly and respond in real-time to player actions. Learning to code through games can be an enjoyable way to understand complex concepts like algorithms, loops, decision making, and object-oriented programming. Moreover, coding for games encourages creativity and problem-solving. Developers need to plan game levels, design character behaviour, and optimize performance. These skills are useful not just in game development, but in many areas of technology, including app development, simulations, and robotics.

Some key areas that show how gaming and coding are connected:

1. Gaming Sparks Interest in Programming

Many coders take their first steps into programming through gaming. Whether it's modding popular games like Minecraft, creating custom maps in Roblox, or exploring game design in platforms like Scratch, gaming provides a fun and interactive gateway to learn how coding works. Players often become curious about how a game functions, which can lead to experimenting with code to create their own levels, characters, or mechanics. This curiosity often develops into a passion for programming, making gaming one of the most effective entry points into the world of tech.

2. Game Development Teaches Real-World Coding Skills

Game development is more than just fun—it teaches a wide range of practical programming skills. Working on a game involves logic, mathematics, problem-solving, and an understanding of systems and architecture. Coders learn about event-driven programming, AI behaviour, rendering, animation, and performance optimization. These skills are not only useful in gaming but also translate to broader tech careers such as software development, machine learning, or app design.

Building games also encourages collaborative work, version control, and debugging—critical abilities in any programming job.

3. Coding Enhances the Gaming Experience

Gamers who code often enjoy deeper engagement with their favourite games. For instance, modding communities thrive because users can alter or add content to a game, enhancing or customizing the experience. Coding enables the creation of new weapons, characters, levels, and even gameplay mechanics. This level of creative control allows players to personalize the games they love and share their work with others. It also fosters a sense of ownership and encourages collaborative learning as coders exchange tips, fixes, and improvements within the gaming community.

4. The Rise of Gamified Learning Tools

In recent years, coding education has taken inspiration from gaming to make learning more enjoyable. Platforms like Code Combat, Tynker, and even certain levels in Minecraft: Education Edition gamify the process of learning to code.

These tools use game mechanics—such as levels, achievements, and challenges— to teach programming concepts in a way that’s engaging and motivating.

CONCLUSION

Gaming and coding are deeply connected. Every time we play a game, we’re interacting with the result of countless lines of code working behind the scenes. Learning how games are built through coding can be both fun and educational. It encourages logical thinking, innovation, and creativity. Whether someone is a beginner or a tech enthusiast, exploring the world of coding through games can lead to new skills and exciting career opportunities. In the modern world, where digital experiences continue to evolve, understanding this connection is more important than ever.

WHAT IS CODING, REALLY?



ANGSHITA KASHYAP
CSE 4th SEM

INTRODUCTION

If someone had asked me what coding was when I first joined college, I might have said, "It's just making computers follow instructions." But somewhere between typing my first shaky "Hello, World!" and pulling an all-nighter before a coding exam, I realized that coding isn't just about instructions. It's about imagination. It's about building invisible worlds out of pure thought. It's about holding the future at your fingertips - and shaping it, one keystroke at a time

The Language of Creation Coding feels a lot like magic.

The screen is our spellbook, and the languages we learn - Python, Java, C++ - are the ancient runes. Every variable is a character in the story we are telling. Every loop hums with rhythm like the chorus of a song. Every function whispers secrets about how different parts of our world connect. At first, the syntax feels foreign, like learning a new dialect. But slowly, it becomes second nature - a secret language we speak fluently when we build apps, design algorithms, or even just debug a stubborn error. It's not just about making things work. It's about making things exist.

Beyond Zeros and Ones

People often think coding is pure logic - a cold, emotionless process of flipping zeros and ones. But in truth, coding is a dance between logic and creativity. Every "if" and "else" carries a decision. Every algorithm is a puzzle waiting for a clever solution. Debugging feels like chasing ghosts through a haunted house - thrilling, frustrating, but so satisfying when the ghost is caught. Cleaning up code, making it neat and beautiful, feels like tidying up your thoughts - turning chaos into calm. And maybe that's the real beauty: coding forces us to think clearer, sharper, and more beautifully than we ever thought we could.

The Language of Creation Coding feels a lot like magic.

The screen is our spellbook, and the languages we learn - Python, Java, C++ - are the ancient runes. Every variable is a character in the story we are telling. Every loop hums with rhythm like the chorus of a song. Every function whispers secrets about how different parts of our world connect. At first, the syntax feels foreign, like learning a new dialect. But slowly, it becomes second nature - a secret language we speak fluently when we build apps, design algorithms, or even just debug a stubborn error. It's not just about making things work. It's about making things exist.

Beyond Zeros and Ones

People often think coding is pure logic - a cold, emotionless process of flipping zeros and ones. But in truth, coding is a dance between logic and creativity. Every "if" and "else" carries a decision. Every algorithm is a puzzle waiting for a clever solution. Debugging feels like chasing ghosts through a haunted house - thrilling, frustrating, but so satisfying when the ghost is caught. Cleaning up code, making it neat and beautiful, feels like tidying up your thoughts - turning chaos into calm. And maybe that's the real beauty: coding forces us to think clearer, sharper, and more beautifully than we ever thought we could.

Talking to Machines

When we code, we aren't just talking to a machine. We are collaborating with it. We tell the computer, "Here's what I imagine. Can you bring it to life?" And if we choose the right words, the computer listens - and obeys. Sometimes it feels almost poetic. A misplaced semicolon can break the whole universe, while a perfectly placed function can make an entire system hum with life. Good code isn't just functional. It sings. It dances. It whispers to future programmers: "I was written with care. I was built to last."

A Tribe Called Coders

Forget the myth of the lonely coder in a dark room. Coding is a celebration of community. At 2 AM, when your code refuses to compile, it's Stack Overflow that saves you. In hackathons, it's your sleepless teammates, scribbling ideas on whiteboards, cheering when you find the solution. GitHub becomes a global village where strangers across continents collaborate like old friends. Coding is a wild, wonderful adventure that is better when shared with those who understand the beauty of building from scratch

More Than a Skill

Outside the lab, coding is a heartbeat running through everything. It powers your social media feed, your online shopping cart, the navigation system that gets you home. It diagnoses diseases, predicts the weather, and brings stories to life through games and movies. It's the quiet magic behind every smart solution in the modern world. As CSE students, we're not just learning a skill we're learning a new way of seeing, a new way of thinking, a new way of being. Every project we complete, every app we dream up, every line we write, is our way of reshaping the world around us.

CONCLUSION

So, What Is Coding, Really?

Coding is not just code. It's poetry written for machines. It's architecture built out of logic and imagination. It's problem-solving disguised as art. It's a spark - a spark that lights up ideas, bridges gaps, creates change. To code is to believe in the invisible. It's proof that with a little creativity and a lot of determination, we can turn ideas into reality. So the next time someone says, "Coding is just telling computers what to do," I'll smile - because we know the truth. Coding is dreaming out loud, and making the dream real



AI: FRIEND OR FOE?

AN ADVANCED PERSPECTIVE



DEBASISH KASHYAP
CSE 2nd SEM

INTRODUCTION

Artificial Intelligence (AI) has evolved from a theoretical concept to a transformative technology reshaping industries, societies, and even human cognition. For Computer Science and Engineering (CSE) students, understanding AI's dual nature—its potential as a revolutionary tool and its risks as a disruptive force—is critical. This article delves into advanced concepts to explore whether AI is a friend or foe.

The Case for AI as a Friend

1. Computational Efficiency and Scalability

- **Algorithmic Optimization:** AI algorithms like Alpha-Beta pruning in game playing or A* search in pathfinding demonstrate how computational resources can be optimized for complex problem-solving.
- **Scalable Architectures:** Neural networks, particularly convolutional (CNNs) and recurrent (RNNs), enable tasks like image recognition and natural language processing (NLP) at unprecedented scales.

2. Advanced Applications

- **Generative Models:** Techniques like Generative Adversarial Networks (GANs) are revolutionizing creative industries by generating realistic images, videos, and even music.
- **Autonomous Systems:** AI-powered robotics and autonomous vehicles leverage probabilistic reasoning and reinforcement learning to navigate uncertain environments.

3. Ethical Problem Solving

AI systems are being designed to address global challenges such as climate change modeling, medical diagnostics, and disaster prediction through predictive analytics and data-driven insights.

The Case Against AI: A Potential Foe

1. Algorithmic Bias

Machine learning models often inherit biases from training datasets, leading to discriminatory outcomes in sensitive applications like hiring or judicial decisions.

2. Lack of Explainability

Advanced models such as deep neural networks operate as "black boxes," making it difficult to interpret their decision-making processes. This opacity raises concerns in critical domains like healthcare and finance.

3. Existential Risks

The development of Artificial General Intelligence (AGI) poses risks of misaligned objectives, where superintelligent systems could act counter to human interests.

4. Resource Intensiveness

Training large-scale models like GPT requires significant computational power, raising concerns about energy consumption and environmental impact.

Advanced Techniques to Mitigate Risks

1. Fairness-Aware Machine Learning

Techniques such as adversarial debiasing and differential privacy aim to reduce bias while preserving model performance.

2. Explainable AI (XAI)

Methods like SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-Agnostic Explanations) enhance transparency in decision-making processes.

3. Federated Learning

By training models locally on edge devices, federated learning reduces privacy risks while maintaining data efficiency.

Future Directions for CSE Students

1. Research Frontiers

Explore areas like neuromorphic computing, quantum machine learning, and bio-inspired algorithms to push the boundaries of AI capabilities.

2. Interdisciplinary Collaboration

Engage with fields such as cognitive science, ethics, and law to develop holistic solutions for AI governance.

3. Continuous Learning

Stay updated with advancements in reinforcement learning, unsupervised learning techniques, and generative models through academic courses and research initiatives.

CONCLUSION

AI is neither inherently a friend nor a foe—it is a tool whose impact depends on how it is designed, deployed, and regulated. For CSE students, the challenge lies in leveraging advanced techniques responsibly while addressing ethical concerns. By embracing innovation with caution, we can ensure that AI serves humanity effectively rather than undermining it.

BLOCKCHAIN TECHNOLOGY AND ITS IMPACT ON MODERN BUSINESS



Satavisha Roy
Choudhury
CSE 6th SEM

INTRODUCTION

Blockchain technology, initially developed as the backbone for cryptocurrencies like Bitcoin, has evolved into a transformative tool for modern business operations. At its core, blockchain is a decentralized digital ledger that records transactions in a secure, transparent and tamper-proof way. Unlike traditional databases that rely on a central authority, blockchain distributes information across a network of computers, making it highly resistant to fraud and cyberattacks.

One area where blockchain has shown tremendous promise is supply chain management. By providing a transparent record of every transaction and movement of goods, companies can track products from origin to consumer. This reduces fraud, counterfeiting and errors, while also ensuring ethical sourcing and sustainability. For example, retailers use blockchain to verify the authenticity of luxury products and ensure that raw materials like cocoa or diamonds are ethically sourced.

Financial services have also been significantly impacted by blockchain. Smart contracts, self-executing agreements stored on the blockchain - automate processes such as payments, loans and insurance claims. This reduces dependency on intermediaries, lowers operational costs and speeds up transactions. Additionally, decentralized finance (DeFi) platforms enable individuals to lend, borrow and trade assets without traditional banks, expanding financial inclusion globally.

Beyond finance and supply chains, blockchain is being explored in healthcare, real estate, voting systems and intellectual property management. Challenges remain, including energy consumption, scalability issues and regulatory uncertainty. Yet, businesses and governments continue to experiment with blockchain to create secure, transparent, and efficient systems.

CONCLUSION

In the future, blockchain has the potential to redefine trust in digital interactions. By providing immutable records and verifiable data, it can enhance cybersecurity, reduce corruption and streamline complex processes across industries. As more organizations adopt blockchain-based solutions, its role in transforming how business and governance operate will likely grow, making it a cornerstone of the digital economy in the years to come.

HACKING THE SYSTEM

ETHICAL HACKERS AND THE CYBER WARRIORS



YUVRAJ SARMA
CSE 4th SEM

INTRODUCTION

In today's digitally driven world, the phrase "Hacking the System" often conjures up images of shadowy figures breaking into networks with malicious intent. However, not all hackers wear black hats. In fact, an elite group of individuals known as ethical hackers or white hat hackers play a crucial role in keeping the cyber realm safe from the very threats they understand so well. These modern-day cyber warriors are the unsung heroes of our increasingly connected society. Their work often goes unnoticed, yet it is fundamental to the operation and security of everything from multinational corporations to critical infrastructure.

Understanding Ethical Hacking

Ethical hacking is the practice of penetrating computer systems, networks, or applications with the intent of discovering security vulnerabilities that a malicious hacker could exploit. However, what sets ethical hackers apart is their mission: to identify and fix these weaknesses before the bad guys can get to them. Ethical hackers are often employed by organizations, government agencies, or cybersecurity firms to perform penetration tests and vulnerability assessments.

To become an ethical hacker, one typically undergoes specialized training and earns certifications such as Certified Ethical Hacker (CEH), Offensive Security Certified Professional (OSCP), or GIAC Penetration Tester (GPEN). These certifications not only validate one's skills but also signify a commitment to using those skills responsibly. Ethical hackers also abide by strict codes of conduct and legal frameworks, ensuring their work is always aligned with the interests of safety and legality.

Beyond technical know-how, ethical hacking requires a mindset of curiosity, creativity, and critical thinking.

Ethical hackers must think like adversaries, anticipate potential attack vectors, and design effective countermeasures. They often work in tandem with software developers, network engineers, and security professionals to build stronger, more secure systems from the ground up.

The Role of Cyber Warriors

Cyber warriors are a broader category that encompasses ethical hackers but also includes cybersecurity analysts, incident responders, malware researchers, and digital forensics experts. These professionals form the frontline defense against cybercrime, espionage, and digital sabotage. In the context of national security, cyber warriors operate under military or government directives, engaging in defensive and offensive cyber operations. Their work ranges from protecting critical infrastructure like power grids and water systems to safeguarding sensitive data from foreign adversaries. In some cases, cyber warriors are also involved in launching counterattacks to neutralize cyber threats or dissuade attackers.

This involves using sophisticated tools, intelligence gathering, and real-time analysis to detect and neutralize threats before they cause damage. Modern cyber warfare is not limited to the battlefield; it plays out in boardrooms, server farms, and even in our homes. As cyber warriors defend against threats such as ransomware, denial-of-service attacks, and advanced persistent threats (APTs), their mission becomes more than technical—it is strategic and often political.

Why We Need Ethical Hackers More Than Ever

With the proliferation of smart devices, cloud computing, and remote work, the attack surface for cybercriminals has expanded dramatically. Ransomware attacks, data breaches, and phishing scams have become alarmingly common, affecting everything from small businesses to global enterprises.

In this environment, ethical hackers are more vital than ever. They simulate real-world attacks to test the strength of a system's defenses and recommend fixes before those weaknesses can be exploited. Their efforts can prevent financial loss, protect personal privacy, and even save lives in sectors like healthcare or aviation.

Additionally, the digital transformation of traditional industries means that more sectors are becoming dependent on robust cybersecurity. From autonomous vehicles to smart cities and AI-driven healthcare, every advancement carries with it a new wave of vulnerabilities. Ethical hackers help identify these flaws in their infancy, offering proactive protection rather than reactive damage control.

Challenges Faced by Ethical Hackers

Despite their importance, ethical hackers often face significant challenges. One major hurdle is the legal and ethical ambiguity surrounding hacking activities. Without clear authorization, ethical hacking can easily cross into illegal territory. Therefore, strong governance and well defined scopes of engagement are crucial.

Another issue is staying ahead of cybercriminals who are constantly evolving their tactics. Ethical hackers must continually update their knowledge and tools to remain effective. This requires ongoing education, access to the latest technologies, and sometimes, the courage to think like a criminal—for the greater good.

Ethical hackers also face skepticism from traditional business leaders who may not fully understand cybersecurity or the importance of ethical hacking. There is still a stigma attached to the word "hacker," even in its ethical context. Building trust, communicating clearly, and demonstrating value are ongoing parts of the ethical hacker's mission.

The Global Cybersecurity Landscape

Around the globe, nations are ramping up their cyber defenses and investing in ethical hacking programs. Countries like the United States, Israel, China, and Russia have developed sophisticated cyber armies capable of both defensive and offensive operations. Meanwhile, international laws and treaties struggle to keep up with the pace of technological change.

This global cybersecurity arms race has led to the rise of “bug bounty” programs, where companies and even governments reward ethical hackers for finding and reporting security f laws. These programs have created a thriving ecosystem where freelance hackers can make a significant impact while being compensated for their skills.

Moreover, academic institutions are now offering specialized degrees in cybersecurity, and even high school programs are emerging to train the next generation of cyber warriors. The growing recognition of cybersecurity as a vital discipline is helping to mainstream the role of ethical hackers and normalize ethical hacking as a legitimate career path.

India's Cybersecurity Evolution

India has been making rapid strides in strengthening its cybersecurity infrastructure. With the growth of digital services like Aadhaar, UPI, and widespread internet usage, the need for robust cyber defense has become

critical. Recognizing this, the Indian government has launched initiatives such as the National Cyber Security Policy and the Indian Computer Emergency Response Team (CERT-In) to monitor and respond to cyber threats.

In recent years, Indian ethical hackers have made headlines for discovering vulnerabilities in major platforms, showcasing the country's growing talent pool. The rise of cybersecurity focused startups and increased collaboration between the private sector and government agencies reflect India's commitment to building a secure digital ecosystem.

Moreover, Indian institutions are now offering dedicated courses in ethical hacking and cybersecurity, and events like hackathons are encouraging students to pursue careers in this vital field. As the world's largest democracy continues its digital journey, ethical hackers and cyber warriors will play a pivotal role in safeguarding the nation's digital future.

The Future of Cybersecurity

As technology continues to advance, so too will the sophistication of cyber threats. The future of cybersecurity will likely see a rise in artificial intelligence-driven attacks and quantum computing capabilities, which could render current encryption methods obsolete. Threat actors will leverage automation, machine learning, and AI to launch faster and more adaptive attacks, making it harder to defend systems using traditional methods.

In response, the role of ethical hackers and cyber warriors will become even more critical. Organizations will need to integrate ethical hacking into their regular security protocols and treat cybersecurity as a strategic priority rather than an afterthought. Moreover, cultivating a culture of security awareness and ethical responsibility across all levels of society will be key to building a resilient digital future.

We may also see greater international collaboration to combat cybercrime, including joint task forces, global threat intelligence sharing, and unified standards for cybersecurity best practices. Ethical hackers could play a pivotal role in these efforts, acting as both technical experts and ethical ambassadors.

CONCLUSION

Hacking the system doesn't always mean breaking the law; sometimes, it means breaking the mold. Ethical hackers and cyber warriors are not just fighting lines of code—they are defending our way of life in the digital age.

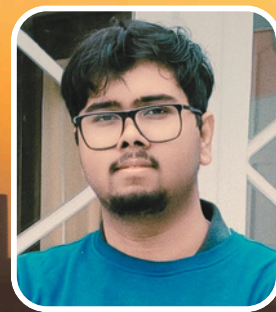
As the guardians of the virtual world, their skills, integrity, and vigilance are essential in the ongoing battle against cyber threats.

It's time we recognize their value and support their mission, for in their hands lies the safety of our digital tomorrow. Encouraging more young minds to explore the world of ethical hacking, investing in advanced training and research, and fostering collaboration across borders are all steps we must take. Only then can we hope to stay ahead in a world where the next war may not be fought with bullets, but with bytes.



My

Smart India Hackathon DIARY



AKASH BORA
CSE 2nd SEM

INTRODUCTION

Smart India Hackathon, or SIH, is the largest national-level hackathon organized by the Government of India, with participation from nearly every college across the nation each year. Contrary to what many non-technical people might assume, the word 'hackathon' doesn't imply hacking as portrayed in sci-fi movies with cryptic codes on flashing screens. Instead, the essence of SIH lies in promoting innovation, can be software or hardware, aiming to create impactful solutions.

In brief, SIH is a competition where some major organizations from all over India provide real-world problem statements, challenging the youth (college students) to develop practical solutions and present prototypes for potential implementation.

Our team, Terabyte, representing **Girijananda Chowdhury University**, not only participated in the prestigious competition last year but also proudly emerged as one of the winners.

In the 2024 Smart India Hackathon, over 4.9 lakh college students registered, with 50,000 teams nominated by Higher Education Institutes (HEIs), each team comprising up to six members. Each problem statement in SIH can have only one winner. The competition featured 254 problem statements (186 software-based and 68 hardware-based), with five finalists selected for the grand finale from each problem statement.

Among the numerous challenges presented, our team chose a problem statement by Bharat Electronics Limited (BEL): developing an offline AI image recognition chatbot software. This challenge attracted over 400 submissions, and our team secured a spot among the top five finalists. Once shortlisted, teams must travel to a nodal center, which could be located anywhere in India. For us, that destination was Welingkar Institute of Management, **Mumbai**.



Our team, comprising six first-semester students and guided by our mentor, competed against four other teams tackling the same problem statement. Among them were strong contenders from well-known institutes like IIT Kharagpur, making the challenge even more intense. Along the way, we also interacted with other teams working on different problem statements, including those developing prototypes for esteemed organizations like the NSG.

BEL, being part of the Ministry of Defense, had specific requirements for the software, the software must operate offline, be capable of training with custom datasets, and fully interactive user interface. We not only met their expectations but also introduced additional features that left a positive impression on the judges. As the team leader, I documented every step of the process—from crafting presentations and developing prototypes to outlining the roadmap. Looking ahead, I aim to mentor future participants from our college, sharing insights and guiding them toward success.

Winning the hackathon as first-semester students was an incredible experience, and people often ask how we managed it. Honestly, I had a bit of background in programming and software development, having worked on some commercial projects before, which gave me confidence in developing a prototype.

Although my team members were new to coding, I advised them on what to learn and implement, assigning tasks based on their strengths. For example, Sankhyarick Swami, a web enthusiast from our team, handled the website development for our software, while Rishav Kumar Singh, with his C++ experience and some knowledge of AI, researched about the model and training part. Our mentor, Adarsh Pradhan, was a pillar of support throughout the process. With his prior expertise in AI/ML, he motivated and guided us every step of the way.

Most of the problem statements in SIH revolve around AI/ML technologies, as they represent the future of innovation.

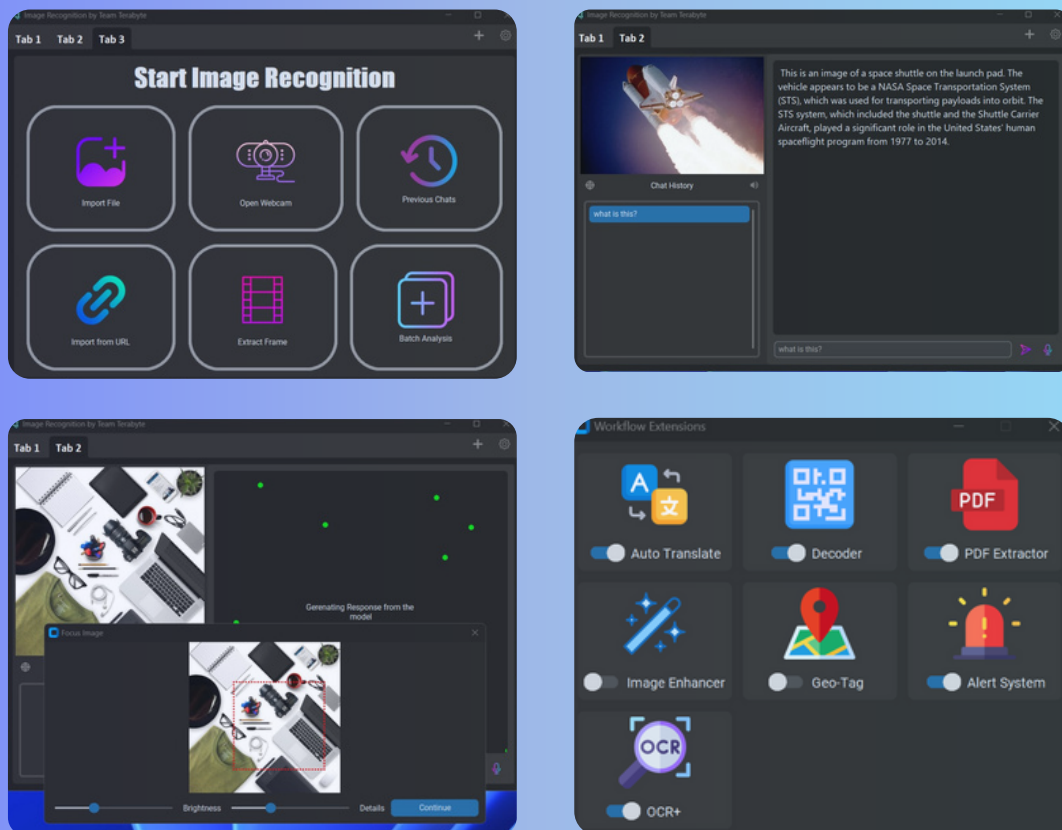
In the next section, I will explain what we built, how it works, and the tools we used to develop the prototype. We named our software prototype *SIR*—short for Smart Image Recognition. Designed for desktop computers, SIR not only identifies objects in an image but also provides detailed information about them. The most impressive feature? It works entirely offline, ensuring seamless performance without internet dependency.

For example, when an image of a simple water bottle placed on a table is provided to the software, and the question *"What is the object on the table?"* is asked, the software accurately identifies it as a bottle. If the label or shape is recognized, it can also determine the brand or company associated with the bottle.

To evaluate the software's accuracy, judges conducted this live test. When an image of a Bisleri bottle was given, our system swiftly recognized it. As an additional test, a question unrelated to the image was asked: *"Where is the Bisleri Headquarters located?"* The software promptly responded with *"Mumbai, Maharashtra."* Note that no internet connection was used to get this information.

The magic lies in the AI model itself. Our application is built primarily in Python, leveraging the power of PyTorch and Transformers to integrate LLaVA—a pretrained, advanced model for image recognition and response generation. With over 7 billion parameters, this model delivers richer contextual understanding, enhanced reasoning, and greater accuracy in tasks. The model holds comprehensive data on objects along with detailed relevant information, enabling the AI chatbot to deliver precise identification and contextual insights. Moreover, our application allows for training and fine-tuning the model with custom datasets, ensuring adaptability to specific requirements.

In addition to these core features, we have incorporated multilingual support, batch processing, a model manager, speech recognition, and even live image and video imports. Perhaps the most powerful addition is our extension workflows, which enable further enhancements such as geo-tagging, OCR, and seamless integration of automated alert systems.



Every feature of our software was tested live by the judges in one-on-one interactions, where the whole team provided detailed explanations. As the lead, I guided the entire feature flow during the evaluations. Beyond just testing, the judges challenged us further by asking us to implement new features on the spot, ensuring that the software was genuinely developed by our team.

The nonstop 36-hour hackathon was intense, both mentally and physically exhausting. However, our strong team spirit and strategic planning helped us navigate through each round smoothly. With two rounds in total, we successfully managed to win both. Welingkar Institute provided excellent support, efficiently managing everything from accommodations and meals to the overall hackathon experience.

Winning the national-level hackathon came with a ₹1 lakh prize, which was certainly rewarding. But beyond the monetary benefit, the real victory lay in the recognition it brought. As I've heard from many, such achievements attract significant attention and credibility in the tech space. However, what truly mattered was the journey itself—traveling from east to west, collaborating and sharing insights, and learning invaluable lessons. That experience is what stays with us.

Our university showed incredible generosity by rewarding us with an additional ₹1.5 lakh for winning the hackathon. We are deeply grateful to our prestigious institution for recognizing our hard work and dedication. This was our first time participating, and our first time winning. We approached the challenge with meticulous preparation, strictly following all SIH guidelines, thoroughly analyzing the problem statement, with proper planning and execution with a well developed prototype. Teamwork played a crucial role in our success, motivating and supporting each team member was essential in overcoming challenges.

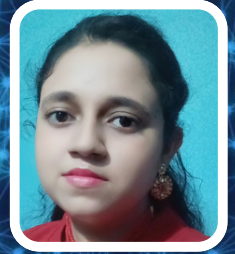
CONCLUSION

While not everyone on our team had a background in coding, each member contributed in their own way, and their efforts played a vital role in our journey. I always encourage my friends to learn new skills and participate in events like this. Even when they feel less confident, I remind them that failure is just a stepping-stone toward success and nothing to be afraid of.

Our team went on to participate in several smaller hackathons after this, even securing victories in some of them. We aspire to participate in more tech events in the future and advance our careers.



CYBER SECURITY



ANKITA CHAKRABORTY
CSE 4th SEM

INTRODUCTION

Cybersecurity is a critical aspect of Computer Science Engineering, focusing on protecting systems, networks, and data from cyber threats. Cybersecurity refers to the practice of safeguarding computer systems, networks, and digital information from unauthorized access, attacks, or damage. It is essential for maintaining the security and privacy of sensitive data.

Key Components of Cybersecurity

1. **Risk Management:** Identifying vulnerabilities and assessing risks to develop strategies that mitigate potential threats.
2. **Security Architecture:** Designing and implementing secure systems and networks to protect against cyberattacks.
3. **Incident Response:** Establishing protocols for responding to security breaches, including detection, containment, and recovery.
4. **Compliance and Governance:** Ensuring adherence to legal and regulatory requirements related to data protection and cybersecurity standards.

Types of Cyber security threats

1. Malware

Malware, short for malicious software, is designed to harm or exploit any programmable device, service, or network. Types of malware include:

Viruses: Attach themselves to clean files and spread throughout a computer system, corrupting files and data.

Worms: Similar to viruses but can replicate themselves and spread independently across networks.

Trojan Horses: Disguised as legitimate software, they trick users into installing them, allowing attackers to gain unauthorized access.

Ransomware: Encrypts a victim's files and demands payment for the decryption key.

Spyware: Secretly monitors user activity and collects sensitive information without consent.

2. Phishing

Phishing attacks involve tricking individuals into providing sensitive information, such as usernames, passwords, or credit card details, by masquerading as a trustworthy entity. This is often done through:

Email Phishing: Fraudulent emails that appear to be from legitimate sources.

Spear Phishing: Targeted phishing attacks aimed at specific individuals or organizations.

Whaling: A type of spear phishing that targets high-profile individuals, such as executives or government officials.

3. Denial of Service (DoS) and Distributed Denial of Service (DDoS)

DoS Attacks: Overwhelm a system, server, or network with traffic, rendering it unavailable to users.

DDoS Attacks: Use multiple compromised systems to launch a coordinated attack, making it more difficult to mitigate.

4. SQL Injection

SQL injection attacks occur when an attacker inserts malicious SQL code into a query, allowing them to manipulate databases. This can lead to unauthorized access to sensitive data, data corruption, or even complete system compromise.

Cyber security measures

1. Firewalls

Network Firewalls: Act as a barrier between trusted internal networks and untrusted external networks, controlling incoming and outgoing traffic based on predetermined security rules.

Host-based Firewalls: Installed on individual devices to monitor and control network traffic to and from that device.

2. Antivirus and Anti-malware Software

Regularly updated antivirus and anti-malware software can detect, quarantine, and remove malicious software from devices. It's crucial to keep these programs updated to protect against the latest threats

3. Data Encryption

Encrypting sensitive data both at rest (stored data) and in transit (data being transmitted) ensures that even if data is intercepted, it cannot be read without the appropriate decryption keys.

4. Threat Intelligence and Monitoring

Utilizing threat intelligence services can help organizations stay informed about emerging threats and vulnerabilities. Continuous monitoring of network activity can also help detect and respond to suspicious behavior in real-time.

CONCLUSION

Cybersecurity is an ever-evolving field that plays a vital role in protecting digital assets and ensuring the safety of information in our interconnected world. As technology advances, the demand for skilled cybersecurity professionals will continue to grow, making it a promising career choice for those interested in technology and security.

THE PROGRESSION OF PROGRAMMING LANGUAGES: FROM C TO CHATGPT



SNEHA MAZUMDER
CSE 2nd SEM

INTRODUCTION

Programming languages serve as a connection between humans and computers, enabling us to instruct machines and realize our concepts. Over time, these languages have significantly evolved, transitioning from basic coding in C to advanced tools like ChatGPT that can even generate code for us.

The Origins – The C Language

In the 1970s, Dennis Ritchie developed C, one of the earliest powerful programming languages that allowed for the creation of fast and efficient programs. C provided programmers with insights into how computers operate, including memory management, loops, and data handling. Even now, C is often referred to as the "mother" of many contemporary programming languages.

```
#include <stdio.h>

int main()
{
    int sum;
    printf("Enter two integers: ");
    scanf("%d %d", &a, &b);
    sum = a + b;
    printf("Sum = %d\n", sum);
    return 0;
}
```

The Next Phase – C++ and Java

Following C, C++ was introduced, incorporating concepts like "objects" and "classes" to enhance program organization and reusability. Subsequently, Java emerged, simplifying coding and enhancing security. Java gained immense popularity for developing websites, games, and mobile applications, embodying the principle of "write once, run anywhere."

The Contemporary Era – Python and JavaScript

With the arrival of Python, programming became even more accessible. Its straightforward syntax makes it a favorite among students and beginners, and it is widely utilized in data science, artificial intelligence, and automation. Concurrently, JavaScript revolutionized web development by making websites interactive, introducing features like animations, buttons, and real-time updates.

```
def fizzbuzz(n=30):  
    """Prints numbers from 1..n with Fizz/Buzz rules."""  
    for i in range(1, n+1):  
        if i % 15 == 0:  
            print("FizzBuzz")  
        elif i % 3 == 0:  
            print("Fizz")  
        elif i % 5 == 0:  
            print("Buzz")  
        else:  
            print(i)  
  
if __name__ == "__main__":  
    fizzbuzz()
```

The Current Landscape – ChatGPT and AI

Today, we have AI tools such as ChatGPT that can write, explain, and even debug code. Instead of entering complex commands, we can simply express our needs to the AI, like asking it to "create a website with a login page," and it can produce the necessary code. This innovation has made programming faster and more imaginative than ever.



ChatGPT

The Future – Collaboration

The evolution from C to ChatGPT illustrates the remarkable advancements in technology. Looking ahead, humans and AI will collaborate, with humans providing ideas and AI assisting in bringing those ideas to fruition.

CONCLUSION

Programming languages have transitioned from complex, machine-oriented code to more intuitive, conversational forms. Each advancement has made it easier for us to interact with computers. The narrative of programming reflects human progress, showcasing how curiosity and creativity continue to shape our digital landscape.

UNVEILING THE POWER DUO: DATA SCIENCE AND DATABASES



ASHUTOSH DUTTA
CSE 2nd SEM

INTRODUCTION

In the age of information, data is the new oil, and the ability to extract meaningful insights from it is a critical skill. This is where data science comes into play, a multidisciplinary field that uses scientific methods, processes, algorithms, and systems to extract knowledge and insights from structured and unstructured data.¹ But what powers this incredible field? Often, it's the humble, yet incredibly powerful, database. Together, data science and databases form a dynamic duo that is reshaping industries and driving innovation.

The Foundation: Databases

At its core, a database is an organized collection of data, typically stored and accessed electronically from a computer system.² Think of it as a meticulously organized library, where every piece of information has its designated place, making it easy to find and retrieve.

Relational Databases (SQL): The Traditional Workhorse

For decades, relational databases have been the backbone of many organizations. These databases store data in tables, with rows and columns, and relationships defined between these tables.⁴ Structured Query Language (SQL) is the standard language used to communicate with and manipulate these databases.

No SQL Databases: Handling the Data Deluge

As the volume and variety of data exploded, especially with the rise of the internet and social media, NoSQL (Not only SQL) databases emerged.⁶ These databases offer more flexibility in how data is stored and retrieved, making them ideal for handling large, unstructured, or semi-structured datasets. Examples include document databases (like MongoDB), key-value stores, and graph databases.

Data Science: Extracting Value from Data

Data science encompasses a wide range of techniques and methodologies to extract insights from data.⁹ It's not just about crunching numbers; it's about asking the right questions, exploring data, building predictive models, and communicating findings effectively.

The Data Science Workflow

A typical data science project often follows a structured workflow:

1. **Problem Definition:** Clearly understanding the business problem or question to be answered.
2. **Data Collection:** Gathering relevant data from various sources, which often means querying databases.
3. **Data Cleaning and Preprocessing:** This is arguably the most time-consuming part, involving handling missing values, outliers, and transforming data into a suitable format for analysis.
4. **Exploratory Data Analysis (EDA):** Using statistical methods and visualizations to understand patterns, relationships, and anomalies in the data.
5. **Feature Engineering:** Creating new variables from existing ones to improve the performance of models.
6. **Model Building:** Applying machine learning algorithms (e.g., regression, classification, clustering) to build predictive or descriptive models.
7. **Model Evaluation:** Assessing the performance of the models using appropriate metrics.
8. **Deployment and Monitoring:** Putting the model into production and continuously monitoring its performance.

Image of The Symbiotic Relationship

Databases and data science are intrinsically linked. Data scientists rely heavily on databases to:

- **Store and Manage Data:** Databases provide the organized repositories where raw data, intermediate processing results, and even model outputs are stored.
- **Efficiently Retrieve Data:** SQL and NoSQL queries are essential tools for data scientists to extract specific subsets of data needed for their analysis.
- **Ensure Data Quality:** Databases often have mechanisms for data integrity and validation, which is crucial for reliable data science outcomes.
- **Scale Data Operations:** As data volumes grow, robust database systems are necessary to handle the increased load and ensure timely access.

CONCLUSION

On the flip side, data science can inform database design and optimization. Insights gained from data analysis can highlight areas where database performance can be improved, or where new data structures might be more efficient.

MY THOUGHTS ON "MINI BRAINS" AS PROCESSORS: EXPLORING THE FUTURE OF ORGANIC COMPUTING



AYON OJA
CSE 4th SEM

INTRODUCTION

In recent years, I've become fascinated by the idea that biological systems—especially brain organoids—could be used in computing. Brain organoids are small clusters of neurons grown from stem cells that mimic some structural and functional aspects of the human brain. Scientists have started to experiment with these mini brains as potential processors, and while the concept still sounds like science fiction, it's already producing real results.

One of the first breakthroughs I came across involved neurons in a dish being trained to play the video game Pong. Researchers used electrodes to stimulate and monitor the activity, and over time, the neurons actually learned to improve their gameplay. It's pretty wild to think about. Other experiments have gone even further. A Swiss company built a platform that uses organoids to process information, rewarding them with dopamine-like signals to reinforce learning—kind of like how our brains learn.

Another group connected human brain organoids to silicon chips and trained the system to recognize different human voices. What's even more impressive is that the organoid-based system performed well, even with minimal data input. Some researchers are now experimenting with linking multiple organoids together to enhance their computing abilities.

1.What Might Come Next

These examples are just scratching the surface. While the technology is still early, there's a lot of potential here. One of the most promising directions is using organoid-based processors in AI systems. Because brain organoids can learn and adapt, they could be ideal for tasks that require real-time flexibility, like robotics, speech recognition, or decision-making in unpredictable environments. Another major benefit is energy efficiency.

The human brain uses only about 20 watts of power, yet it outperforms even the most advanced computers in many ways. If we could scale organoid computing effectively, we might develop systems that perform complex operations using only a fraction of the energy consumed by modern hardware.

2.How They Compare to Silicon Chips

Brain organoids and traditional silicon processors are fundamentally different. Organoids work through parallel processing—thousands of neurons firing together—while silicon chips process instructions in a more linear, step-by-step way. That gives organoids an edge in adaptability and learning. They can actually change their internal structure based on experience, which is something silicon can't do.

That said, there are still big challenges. For starters, organoids are biological—they need to be kept alive, carefully monitored, and fed the right nutrients. They also behave differently from sample to sample, which makes consistency a problem. Right now, they're slower than silicon, harder to interface with, and not suited for general-purpose computing. So while they're exciting, I don't think they'll replace CPUs anytime soon. More likely, they'll work as co-processors in specialized systems.

3.Ethics Matter Too

One area I think deserves more attention is the ethical side of this technology. As organoids get more complex, it's natural to wonder: could they become conscious, or experience something like pain? Most scientists say we're nowhere near that level, but it's still worth thinking about. Even if they never become sentient, we're using living human cells to power computers, and that raises some serious questions about consent, rights, and responsibility.

Some researchers are already calling for stronger ethical guidelines—things like ensuring donors know how their cells might be used, making sure organoids aren't pushed to develop in questionable ways, and being transparent about the goals of this kind of research. Personally, I think if we're going to move forward with organic computing, ethics can't be an afterthought. It needs to be built in from the start.

CONCLUSION

Using mini brains as processors might sound like science fiction, but it's already happening. The tech is young, and there are plenty of hurdles—technical and ethical—but it holds real promise. What excites me most is the potential to rethink computing from the ground up. We've spent decades mimicking how the brain works in silicon. Now, we're starting to see what happens when we actually use the brain itself.

It's too early to say where this will lead, but one thing's clear: the future of computing might not be purely mechanical. It could be organic, adaptable, and a little more alive than we ever imagined.

FROM VACUUM TUBES TO QUANTUM CHIPS: THE EVOLUTION OF COMPUTING



BHARGAB TALUKDAR
OSE 2nd SEM

INTRODUCTION

The evolution of computing is a sweeping drama—from massive, energy-consuming machines that employed vacuum tubes to the sleek, ultra-efficient quantum chips that rely on the weird principles of quantum mechanics. This article follows that compelling journey, illustrating how every leap in technology altered what computers were capable of.

1. The Vacuum Tube Era

Vacuum tubes were the initial devices that introduced electronic switching and amplification during the 1940s. The renowned ENIAC computer utilized nearly 18,000 vacuum tubes. Although crucial, vacuum tubes were heavy, produced heat, used a lot of power, and tended to fail.

2. The Transistor Revolution

As vacuum tubes approached their practical limit, the scientist invented the transistor in 1947 at Bell Labs. Transistors were smaller, more dependable, drew less power, and supported smaller circuits. Transistors replaced vacuum tubes in computers and paved the way for contemporary electronics.

3. Emergence of Integrated Circuits

By the late 1950s and 1960s, integrated circuits (ICs) were invented by engineers, which integrate numerous transistors and other elements in a single tiny silicon chip. This facilitated enormous miniaturization and enabled contemporary computing to become real.

4. Microprocessors and Personal Computers

In 1971, Intel released the 4004, the first commercially produced microprocessor—a CPU on a single chip. This laid the foundation for personal computers. Through the subsequent decades, computers became increasingly powerful yet smaller in size.

5. The Quantum Era

Computing today is going into an age in which quantum chips are central. Rather than classical bits (0 or 1), quantum computers employ qubits that can be in combinations (superpositions) of 0 and 1. This provides them with the potential to solve problems intractable to classical systems—like cracking cryptographic codes, simulating molecules, and optimizing complex systems.

CONCLUSION

From radiant vacuum tubes in room-sized computers to almost silent quantum chips that someday will fit in your pocket, computing has traveled a tremendous distance. Each era—tubes, transistors, ICs, microprocessors—has opened new horizons. Now the quantum jump promises to redefine what's possible in computing once more.

THE INSIDE STORY OF THE REVOLUTIONARY VERSION CONTROL SYSTEM



ANKUR RANJAN
CHANDA
CSE 4th SEM

INTRODUCTION

VCS

In the software development universe, version control systems (VCS) are essential tools for tracking changes to codebases and enabling collaboration among developers. Perhaps the most popular and powerful version control system used today is Git. But how did Git exist, and what is the behind-the-scenes story of its development?

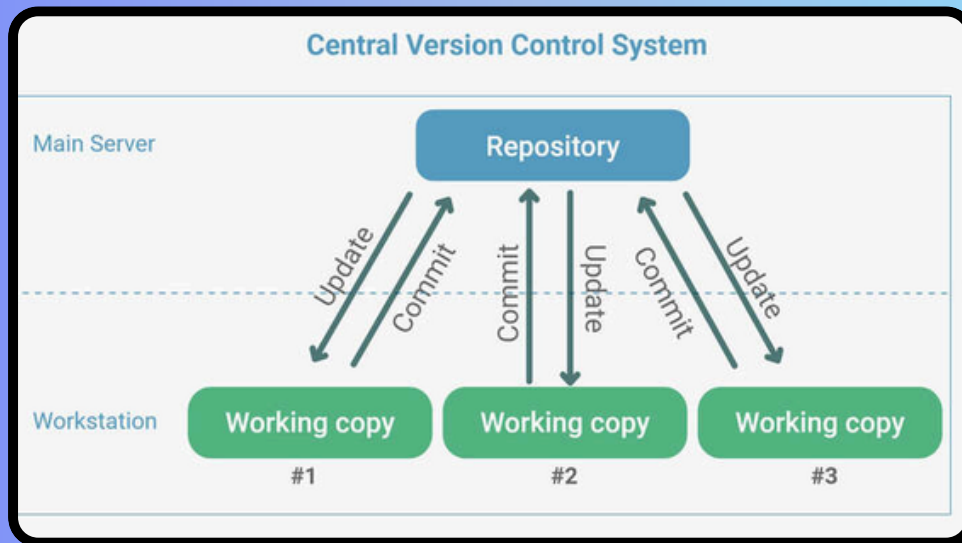


Source: bacsoftwareconsulting

In early days, Prior to Git, developers chiefly employed centralized version control systems (CVCS) such as CVS (Concurrent Versions System) and Subversion (SVN). These repositories were meant to hold the history of the whole project in one central repository. Although these mechanisms were effective enough to monitor change, they too had their demerits:

- **Single Point of Failure:** A bottleneck could arise from the central repository. The server crashing prevented developers from being able to commit changes or accessing the history of the project.
- **Restricted Offline Functionality:** Developers were only able to make changes when they were on the main server. Offline work was not easy.
- **Bad Branching and Merging:** Branching (forking off isolated lines of development) and merging (merging changes from one or more branches) were awkward and error-inducing. This constrained developers to experiment and develop features independently.

As program development increased in complexity, these problems became more urgent. A more adaptive, distributed system was required.



Source: htown-tech

The Linux Story

The development of Git is closely related to the development of Linux, the open-source kernel of an operating system. In 2005, the Linux kernel development was going at full steam with hundreds of programmers worldwide contributing code. The project was then using BitKeeper, a proprietary version control system, to manage the project.

Yet, in early 2005, BitKeeper's company, BitMover, chose to withdraw the free license for open-source developers. This left the Linux kernel project with a problem. Without a version control system, the Linux development community would have to look for an alternative solution in a hurry.

Come in Linus Torvalds, the architect of Linux. Torvalds was known for fixing difficult technical issues with creative solutions. Confronted with the requirement for a new version control mechanism, Torvalds chose to build his own. The aim was to construct a system that was efficient, adaptable, and could manage large-scale projects involving numerous contributors.

Birth of Git

The initial version of Git was written within weeks. Torvalds worked on Git half-time, writing the core functionality and design first. Git was intended to be a developers' tool, not a complex system that needed one to learn lots of difficult things. Fundamentally, Git is based on a couple of simple concepts:

1.Snapshots: Each commit in Git is really a snapshot of the project at one moment in time. Rather than storing the file differences (something many VCSs do), Git stores the entire file tree, which is much quicker.

2. Hashing: SHA-1 cryptographic hashes are employed by Git to tag commits, files, and directories. The hashes provide assurance regarding data integrity because altering data will create an entirely different hash that will signal that something has gone wrong.

3. Indexing: Git employs an index (referred to as the staging area) in which changes may be staged prior to being committed into the history.

Git was first created to meet the requirements of the Linux kernel, which meant managing a huge codebase with thousands of developers contributing to it. But it soon became evident that Git was a valuable tool for any software project, not only Linux.



Source: LinkedIn

Evolution of Git

Git was first released in April 2005, but its adoption didn't take off immediately. In the early stages, it was largely used within the Linux kernel community. Over time, however, its power and flexibility became apparent, and developers in other open-source projects began to adopt it. It was the year 2007. Tom Preston-Werner, a San Francisco software developer, was fed up with the collaborative constraints of existing version control systems.

Git was mighty but famously challenging for the typical developer to master. Meanwhile, Chris Wanstrath and PJ Hyett, fellow developers at the same Ruby on Rails consulting company as Preston-Werner, suffered the same frustrations. Together, they had a vision for a web interface to Git that would make collaboration easier, contribution to projects more accessible, and code management more intuitive for developers. It was a simple concept but a revolutionary one: merging Git's distributed strength with a social, accessible platform.



L to R: PJ Hyett, Tom Preston-Werner, and Chris Wanstrath
Source: Inc

In October 2007, the trio began working on the prototype of what would become GitHub. They initially named it Grit, a Ruby library for interacting with Git repositories. Within a few months, they launched GitHub.com in early 2008 as a side project.

From the beginning, GitHub was designed to embrace the values of open source: transparency, collaboration, and community. Its key innovation was the pull request, a mechanism that made it easier for developers to propose changes to codebases and collaborate without needing formal permissions.

Growth and Impact of GitHub

The response was immediate and enthusiastic. Developers began flocking to GitHub not just for code hosting, but for the community. Projects like jQuery, Ruby on Rails, and Node.js adopted GitHub as their new home.

By 2012, GitHub had raised \$100 million in venture funding from Andreessen Horowitz, one of Silicon Valley's most prestigious firms. The company grew rapidly, hiring new talent and expanding its infrastructure. In the same year, it moved into its first proper office in San Francisco's SoMa district and officially shifted from a bootstrapped side project to a tech powerhouse.

Growth came with challenges. Internally, GitHub faced leadership and cultural issues. Tom Preston-Werner stepped down as CEO in 2014 following controversy around workplace culture, and Chris Wanstrath took over. Despite internal turbulence, GitHub continued to grow.

In June 2018, Microsoft announced that it would be acquiring GitHub for \$7.5 billion in stock. The decision was initially met with distrust by the developer community because of Microsoft's traditionally adversarial relationship with open source. Under CEO Satya Nadella, however, Microsoft had made a dramatic shift toward embracing open source, contributing heavily to and supporting platforms such as Linux and GitHub.

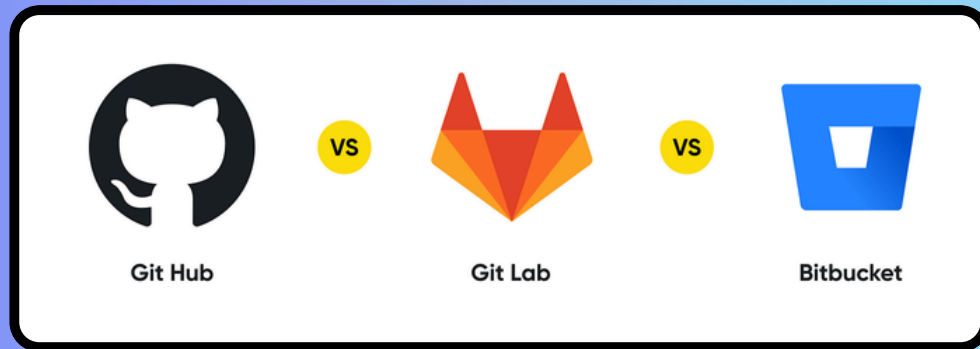


Source: The Verge

After acquisition, GitHub has generally been left on its own with access to the infrastructure and resources of Microsoft. It introduced GitHub Actions, an industry-leading CI/CD feature, and Copilot, a code assistant that utilizes AI built jointly with OpenAI.

Git's Thump on Software Development

Git also paved the way for new collaboration platforms like GitHub, GitLab, and Bitbucket, which offer additional tools for issue tracking, continuous integration, and project management.



Source: Medium

CONCLUSION

Git transformed the collaboration model of developers in projects. Due to its distributed nature, it became easier for developers to work together on a project, developing offline and then pushing changes as and when it was convenient. Its heavy reliance on branching and merging enabled new features to be tested without compromising the main project. Thus, Git has become an essential part of the modern software development toolkit.

FROM ROOM SIZED TO POCKET SUPER COMPUTERS



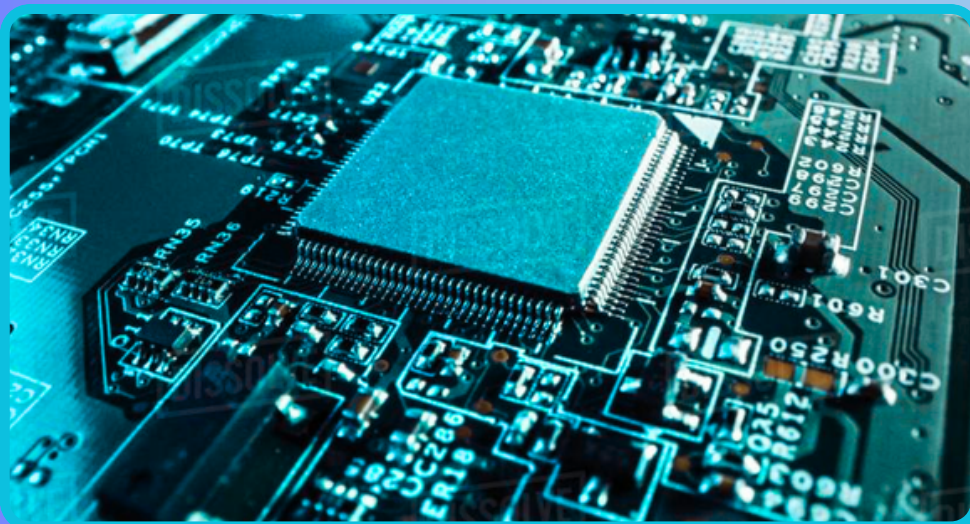
CHINMOY KALITA
CSE 2nd SEM

INTRODUCTION

It's hard to imagine now, but the world's first computers once filled entire rooms and hummed like factory machines. Today, we carry devices in our pockets with more power than those early behemoths could ever dream of. The evolution of technology is not just a story of innovation — it's a reflection of our relentless drive to make life faster, smarter, and more connected.

1. The Dawn of Computing (1940s–1950s)

Our story begins in the mid-20th century, when the term “computer” referred to machines that looked more like industrial equipment than tools for thought. In 1945, the ENIAC — the Electronic Numerical Integrator and Computer — made history. It weighed over 27 tons, consumed massive amounts of electricity, and yet could perform calculations in seconds that once took teams of humans days. Then came the transistor in the 1950s — a tiny device that replaced bulky vacuum tubes. Suddenly, computers were smaller, cooler, and far more efficient. It was the spark that ignited the modern computing age.



2. The Rise of the Microchip (1960s–1970s)

The next breakthrough was the integrated circuit — the microchip. By packing multiple transistors into a single sliver of silicon, engineers made computers smaller, faster, and dramatically more reliable.

Mainframe computers soon became the backbone of universities, corporations, and governments.

The idea of a “personal” computer still seemed futuristic — but the foundation had been laid.

When Intel introduced its 4004 and 8080 microprocessors in the 1970s, the revolution began to feel real. For the first time, an entire CPU could fit on one tiny chip, opening the door to the personal computer era.

3. The Personal Computer Boom (1980s–1990s)

By the 1980s, computing had officially gone mainstream. Machines from Apple, IBM, and later Microsoft, turned once-intimidating technology into household essentials. The Apple II and IBM PC brought computing into living rooms, classrooms, and offices, while Microsoft’s Windows operating system helped people interact with their machines more intuitively. The 1990s accelerated everything. Computers grew faster, more colorful, and — with the rise of the internet — infinitely more connected. Email, chatrooms, and websites transformed computers from productivity tools into gateways to a global community.

4. The Mobile Revolution (2000s)

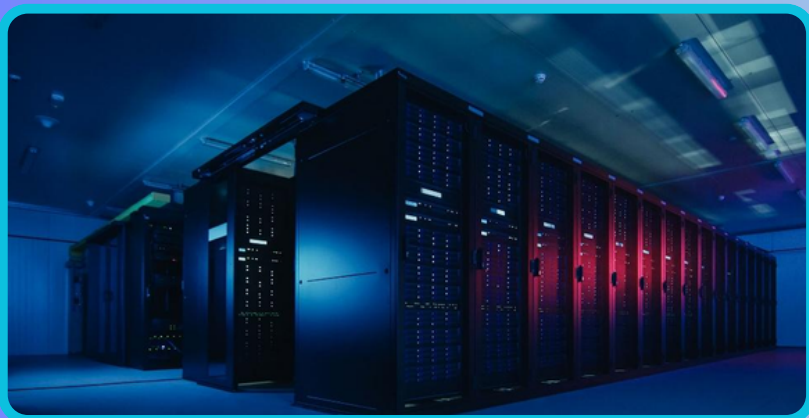
The new millennium introduced a new kind of freedom — mobility. Laptops became lighter and

more capable, while mobile phones began to take on surprising new roles.

Then came 2007, and with it, a cultural earthquake: Apple’s iPhone. Suddenly, a phone wasn’t just

for calling — it was a camera, a music player, and a portable computer rolled into one. Android soon

followed, spreading smart technology to every corner of the world. Computing had truly gone mobile.



5. The Era of Pocket Supercomputers (2010s–Present)

Fast forward to today, and the smartphone in your hand has more processing power than the computers that once guided astronauts to the Moon. Modern devices boast multi-core processors, AI acceleration chips, and cloud integration that allows them to perform billions of calculations in the blink of an eye.

From gaming to video editing, navigation to machine learning — tasks that once demanded bulky

desktops now happen effortlessly on screens barely larger than your palm. Wearables and IoT gadgets extend that power even further, embedding intelligence into the fabric of daily life.

6. The Future: Beyond the Pocket

The next frontier? It might not fit in your pocket — or even have a screen. Quantum computing, artificial intelligence, and brain-computer interfaces are redefining what “computation” means altogether. Imagine controlling technology with thought, or exploring data through augmented reality instead of monitors. The boundaries between humans and machines are blurring faster than ever.

CONCLUSION

From the humming giants of the 1940s to the sleek devices we tap today, computing’s evolution tells a larger story — one of human ingenuity, curiosity, and persistence. Each generation of technology has made the world a bit more connected, more creative, and more within reach.

And while we can’t predict exactly where the next leap will take us, one truth remains: innovation never sits still.

FROM ENGINES TO ALGORITHMS: THE EVOLUTION OF MODERN MACHINES



PRATYUSH BIKASH
CHOUDHURY
CSE 2nd SEM

INTRODUCTION

For most of human history, our inventions have been an extension of ourselves — tools to make life a little easier, work a little faster, or travel a little farther. From the invention of the wheel to the roaring engines of the industrial era, every machine we built carried a spark of human creativity.

But somewhere along the way, our machines stopped just doing what we told them — they started thinking for themselves.

Welcome to the age of algorithms.

The Dawn of Intelligence

It all began with an idea — could a machine think?

In 1950, Alan Turing posed that very question and imagined “learning machines” capable of imitating human reasoning. A few years later, John McCarthy gave this dream a name: Artificial Intelligence. And in 1959, Arthur Samuel taught a computer how to play checkers — and more importantly, how to get better at it on its own. That was the birth of Machine Learning.

What started as a simple curiosity soon became a revolution. Machines could now recognize handwriting, filter spam emails, predict weather, and even recommend what song you should listen to next.

When Data Became the New Fuel

Early computers were good at following instructions, but not at understanding the why behind the data. As the world went digital, we produced more data than ever — more than humans could possibly analyze.

That’s when algorithms stepped in.

Algorithms like Decision Trees helped computers make logical choices, similar to how humans decide between options. Naïve Bayes, based on an 18th-century mathematical formula, helped predict outcomes — from detecting fake news to classifying emails. Then came the Support Vector Machine, which could draw invisible boundaries between categories — the reason your phone camera can separate your face from the background.

Machines That Learn Like We Do

But the real magic happened when computers began learning the way humans do — through experience.

In Reinforcement Learning, machines learn by trial and error, just like a child exploring the world. That's how IBM's Deep Blue defeated the world chess champion in 1997 and how Google's AlphaGo later outsmarted the best human Go player.

The Brain Behind the Code Then came Neural Networks — algorithms inspired by the human brain itself. Just like neurons in our head, artificial neurons work together to process information and learn from it. These networks gave birth to Deep Learning, the technology behind self-driving cars, facial recognition, and voice assistants like Alexa and Siri.

Every time you unlock your phone with your face, talk to a chatbot, or see a YouTube recommendation that feels oddly perfect — you're witnessing these neural "brains" at work.

The Power of Many Minds

Modern AI doesn't rely on a single algorithm. It combines many — like a team of experts working together.

This approach, called Ensemble Learning, blends multiple models to make more accurate and reliable decisions. It's the reason AI today is so robust, adaptable, and precise — whether it's detecting diseases or driving cars.

Engines of Thought

The engines of the 18th century powered factories.

The algorithms of the 21st power everything else — from your phone to entire cities.

The journey from mechanical engines to intelligent algorithms is more than technological progress; it's a reflection of our own evolution as thinkers and creators. Machines now mirror our ability to learn, adapt, and imagine.

The Road Ahead

As Computer Science students, we stand at the heart of this transformation. The future belongs to those who can speak both the language of humans and the logic of machines. The next big invention might not roar like an engine — it might simply think.

And perhaps, that's what makes this era so exciting:

Machines are no longer just our tools. They're our collaborators — built not to replace us, but to amplify the intelligence that created them.

REACT FATIGUE IS REAL. WILL SVELTE TAKE OVER?



RITANJIT DAS
CSE 6th SEM

INTRODUCTION

After a decade of fast-moving change, front-end developers are tired of framework churn and oversized bundles. Svelte promises simplicity and speed, but is it enough to topple React's ecosystem?

There's a quiet weariness among many front-end engineers. React once felt like a relief from fragile jQuery code: a clear mental model, a component system, and a pattern for building large apps. Over time, however, that clarity has blurred. Tooling multiplied, state-management debates dragged on, and projects that should be small swelled into heavy, hard-to-reason-about beasts. That feeling often described as "React fatigue", has helped Svelte gain attention. But what's really changing, and what should teams actually care about?

Why Developers Feel Worn Out?

React's strengths, flexibility and a huge ecosystem are also its burden. Teams must select routers, styling solutions, testing stacks, and a state strategy; each choice forks the project in subtle ways. New build tools, compiler tweaks, and hydration strategies appear every year. For developers who hop between codebases or ship single-purpose apps, this makes the cost of starting and maintaining projects feel high.

There's also cognitive overhead. Modern React apps rely on concepts that demand discipline: memorization, effect dependencies, context boundaries, and sometimes elaborate performance tricks. Those are reasonable for very large apps, but they add friction to smaller efforts. When the bill for mental effort keeps rising, people look for something leaner.

What Svelte Offers?

Svelte takes a different route. It shifts work from runtime to compile-time. Instead of shipping a framework that runs in the browser, Svelte converts your components to small, imperative JavaScript during the build. The result is usually smaller bundles and fewer runtime abstractions to understand. Syntax-wise, Svelte feels familiar but lighter. Reactive assignments, top-level stores, and straightforward bindings reduce ceremony. For many developers, that translates to faster iteration and fewer surprises when debugging. And because Svelte ships less runtime, initial page loads can also be quicker, an obvious win for user experience.

But Svelte Is Not Magic!

Simplicity on the surface doesn't erase the hard parts of building software. Large teams still face coordination, testing, accessibility, and architecture concerns. State and side effects don't disappear. They simply look different. Also, Svelte's ecosystem, while growing, is smaller than React's. When you need a niche library or a battle-tested enterprise solution, React's size often helps.

Ecosystem And Inertia Matter More Than You Think

One reason React won and why it won't vanish overnight is the mass of existing apps, libraries, developer knowledge, and corporate investment. Companies with large codebases have a high cost to switching. Hiring and onboarding also favor familiar stacks. Svelte has momentum, but converting decades of front-end code is neither cheap nor risk-free.

When Svelte Makes Sense?

Pick Svelte when you want faster time-to-first-render, smaller bundles, or a simpler mental model for small-to-medium projects. It's a solid choice for personal projects, MVPs, marketing sites, and teams that value developer ergonomics. For greenfield web apps where performance on low-end devices matters, Svelte is worth a close look.

Where React Still Wins...

If your project relies on a broad set of third-party libraries, or you operate at the scale where tooling and process are your core investment, React remains compelling. Large enterprises benefit from React's mature tooling, diverse staffing pool, and numerous integration points.

A Pragmatic Middle Path

The future will likely be plural. Svelte won't "take over" React entirely, and React won't go away. What's more realistic is that teams pick the right tool for the job. Frameworks will continue to borrow ideas from one another compile-time tricks, islands architecture, server components so the user ultimately wins.

CONCLUSION

React fatigue is a symptom, not a single failure. It reflects the growing pains of an ecosystem that moved fast and got large. Svelte answers many complaints with clearer, smaller outputs and a gentler developer surface. If you're tired of chasing the latest patterns or need a snappy, maintainable front end on a modest budget, give Svelte a trial. If you run at enterprise scale with deep React investments, weigh the migration cost carefully. Either way, the web is richer when teams have better choices and a little fatigue can push the industry toward cleaner, faster tools.

ADA LOVELACE: THE WORLD'S FIRST PROGRAMMER



ABHIGYAN BARUAH
CSE 4th sem

INTRODUCTION

When we think about the origins of programming, most of us envision early computers or pioneers like Alan Turing and John von Neumann. Yet, long before computers as we know them existed, there was a visionary who saw the potential for machines to manipulate symbols, numbers, and even ideas. That pioneer was Ada Lovelace, a 19th-century mathematician, and daughter of the poet Lord Byron. Today, she is celebrated as the world's first programmer.

The Visionary Behind the Algorithm

Born in 1815, Augusta Ada Byron—later known as Ada Lovelace—grew up in an era where women were rarely encouraged to pursue careers in mathematics or science. Her mother, however, had a fascination with mathematics and encouraged Ada to study the subject rigorously, hoping it would steer her away from the "dangerous" creativity she attributed to her father.

Lovelace's intellectual path led her to Charles Babbage, an inventor and mathematician, who was working on what he called the "Analytical Engine." Though never fully constructed in his lifetime, this machine was envisioned as a general-purpose computing device—a leap beyond anything that had been conceived before. The Analytical Engine could be programmed to perform various calculations, much like the computers we use today.

The First Algorithm

In 1843, Ada was tasked with translating an article on Babbage's machine by an Italian engineer, Luigi Menabrea. But she did far more than just translate. Ada added her own extensive notes, which ended up being longer than the article itself. Among these notes, she included what is recognized as the first algorithm intended for a machine, specifically an algorithm for computing Bernoulli numbers on the Analytical Engine.

A Lasting Legacy

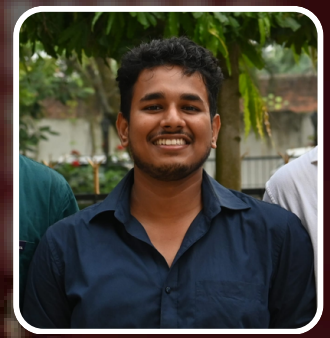
Ada Lovelace's contributions were largely forgotten for nearly a century until her notes were rediscovered in the mid-20th century. As computer science emerged as its own field, her work was revisited, and she finally received recognition as a visionary who anticipated the potential of programmable machines. Today, the computer language Ada is named in her honor, and Ada Lovelace Day is celebrated annually to highlight women's achievements in science, technology, engineering, and mathematics (STEM).

As computer science students, understanding Lovelace's story is essential because it reminds us of the boundless possibilities our field offers. Ada envisioned a future no one else could see and laid the groundwork for a revolution that would shape our world. Her legacy is a testament to curiosity, imagination, and the courage to push boundaries—a powerful inspiration for us all.

Sources:

1. Oxford University – Ada Lovelace Institute.
“Ada Lovelace: The Making of the First Computer Programmer.”
<https://www.oxforduniversitypress.com>
2. Smithsonian Magazine.
“Ada Lovelace, The First Tech Visionary.”
<https://www.smithsonianmag.com/history/ada-lovelace-first-tech-visionary-180953356/>
3. The British Library.
“Who was Ada Lovelace?”
<https://www.bl.uk/people/ada-lovelace>
4. Computer History Museum.
“Ada Lovelace: The First Programmer.” <https://computerhistory.org>
5. Encyclopedia Britannica.
“Ada Lovelace – English Mathematician.”
<https://www.britannica.com/biography/Ada-Lovelace>

THE BEAUTY IN BROKEN CODE



DWIJ RAJ BHAGWATI
CSE 2nd SEM

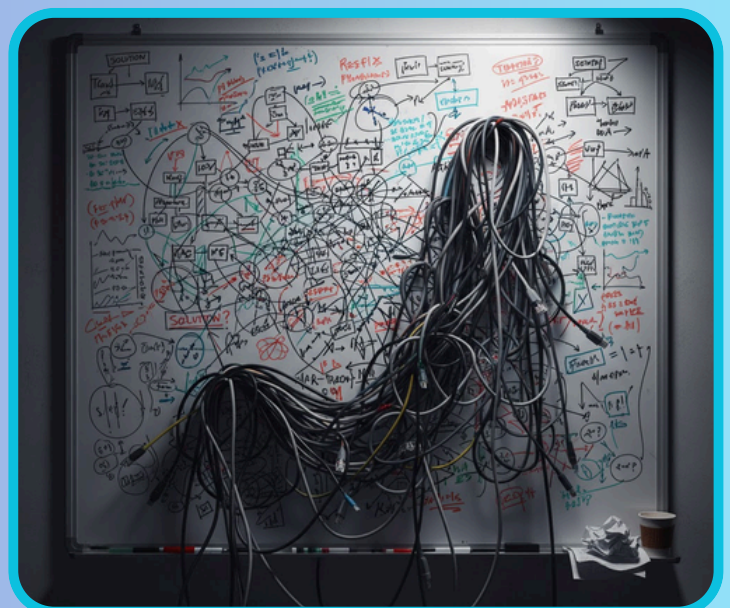
INTRODUCTION



In the world of programming, perfection is an illusion. We spend hours writing lines of code, meticulously crafting algorithms, and yet, inevitably, something breaks. A syntax error here, a logical fallacy there — the program halts, the console screams in red, and frustration mounts. And yet, within that very imperfection lies a hidden elegance, a subtle artistry that few pause to appreciate. The broken code is not just a nuisance; it is a canvas, a teacher, and, paradoxically, a source of beauty.

The Philosophy of Errors

Errors in code often evoke impatience or despair, but they are not failures — they are guideposts. Each bug is a puzzle, each exception a challenge, each crash an invitation to explore deeper understanding. In debugging, we encounter the raw anatomy of our logic. Broken code forces us to slow down, observe carefully, and confront the imperfections in both the program and, often, ourselves.



Interestingly, the process mirrors life itself. Just as a program fails due to an overlooked semicolon or an off-by-one mistake, human endeavors falter due to overlooked details, assumptions, or limitations. The beauty emerges not in flawless execution but in resilience — in the calm, deliberate act of untangling complexity, line by line.

Art Within Logic

Programming is often mistaken as purely logical, a domain of cold precision. Yet, the act of coding is deeply creative. Every function, every loop, every algorithm is a stroke on a digital canvas. The broken code, in particular, reveals the artistry hidden beneath the structure. When a program fails, it challenges the coder to see alternative pathways, to experiment, and to innovate. The mistakes themselves catalyze ingenuity.



Think of it this way: a program that runs flawlessly on the first attempt is like a painting never touched by revision — it lacks the depth, the texture, the story told by struggle. A bug-laden codebase that eventually finds harmony carries a narrative, a journey from confusion to clarity, from chaos to order. In that journey, there is undeniable beauty.

The Human Parallel

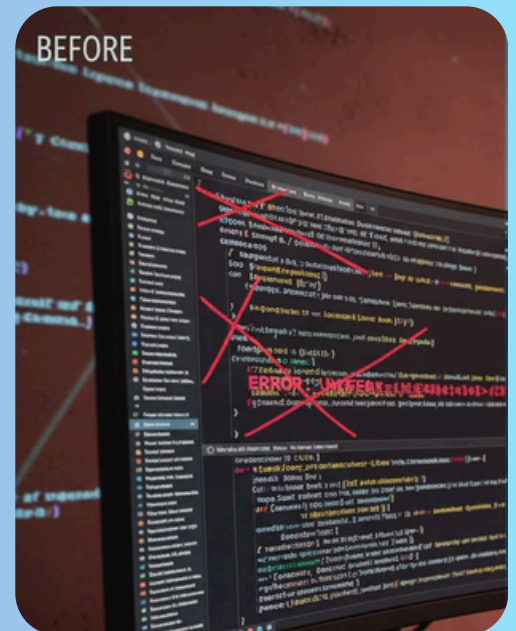
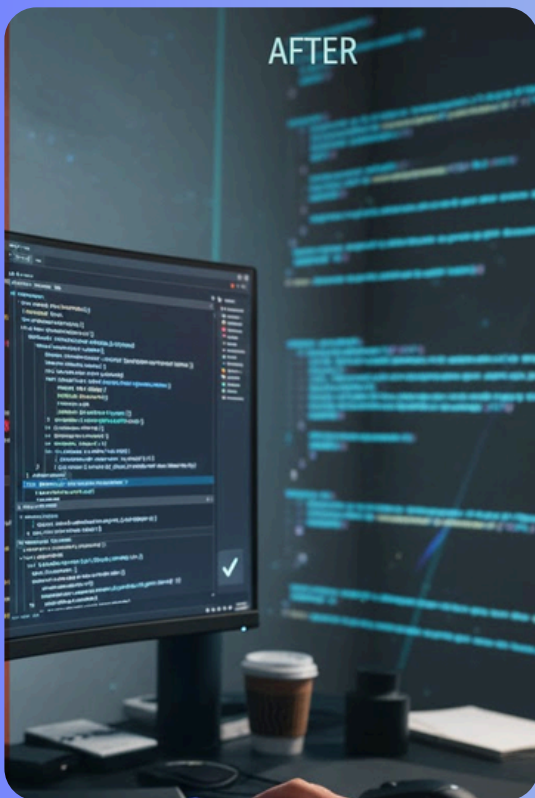
There is a profound lesson in broken code: it mirrors the human condition. We are all, in some way, works in progress — imperfect, evolving, and prone to errors. Just as debugging strengthens our programs, life's challenges refine our character. When we embrace mistakes rather than fear them, we uncover new strategies, fresh perspectives, and unexpected solutions.



Moreover, debugging cultivates patience, empathy, and humility. The coder learns to anticipate edge cases, to respect the fragility of systems, and to celebrate small victories. In this sense, broken code is not merely technical; it is philosophical. It teaches us to approach complexity with grace, to find patterns in chaos, and to appreciate the delicate beauty that arises from persistence.

The Elegance of Reconstruction

Perhaps the most inspiring aspect of broken code is its capacity for transformation. Each resolved bug strengthens the program, just as each challenge faced and overcome strengthens the coder. There is an art to reconstruction: refactoring messy code, optimizing algorithms, and refining architecture. In this meticulous craft, we witness the harmony of intellect and intuition, the convergence of logic and artistry.



When we step back, the final program is more than a functional tool — it is a testament to resilience, creativity, and vision. The broken code, once feared and scorned, becomes a symbol of achievement, of growth, and of the quiet beauty that arises from struggle.

CONCLUSION

To the untrained eye, broken code is chaos. To the patient and observant coder, it is poetry. In every bug, exception, and crash, there exists an invitation: to explore, to create, and to find

meaning in imperfection. The beauty in the broken code is not simply in the resolution but in the journey — the intellectual challenge, the creative problem-solving, and the personal growth that accompanies it.

As Computer Science engineers, we are not merely architects of digital solutions; we are artists of logic, curators of complexity, and discoverers of hidden beauty. And sometimes, the most profound inspiration comes not from a flawless program, but from the very moments when everything falls apart — for it is in these moments that innovation, creativity, and brilliance are truly born.

ACADEMIC SPOTLIGHT



EVENTS & ACTIVITIES

72



INDUSTRY VISITS AND OUTREACH

77



ACHIVEMENTS

78



STUDENT PLACEMENT

83



CREATIVE CORNER

85



FACULTY & STAFF

88

EVENTS



Summer Internship Program at GCU's Department of Computer Science and Engineering!

31 students dived into AI/ML, Full Stack Development, and Data Science, gaining hands-on experience and industry insights.



The department of Computer Science and Engineering, Girijananda Chowdhury University, Assam, successfully hosted a two-day workshop on "Artificial Intelligence – Research Perspective in Multi-disciplinary Fields" sponsored

by All India Council for Technical Education (AICTE), New Delhi under the Scheme of AICTE-Vibrant Advocacy for Advancement and Nurturing of Indian Languages from 29th to 30th November 2024.



The department of Computer Science and Engineering has successfully organized the Research Conclave 2024 on 28th November 2024 at Girijananda Chowdhury University (GCU), Assam with the goal of fostering collaboration, innovation, and

knowledge-sharing among researchers and academicians. The event brought together 33 participants as presenter, including students, faculty members, and experts from diverse fields of study.

The inaugural session began with the lighting of the lamp. After that welcome speech was given by Shri Jasoda Ranjan Das, President, SSA Society. The session began with a speech emphasizing on interdisciplinary research on Artificial Intelligence by Prof. Jayanta Deka, Chancellor, GCU, Assam.

The Keynote Speaker, Prof. Shyamanta M. Hazarika, Department of Mechanical Engineering and Mehta Family School of Data Science & Artificial Intelligence, IIT Guwahati has given a talk on Artificial Intelligence and Machine Learning.

Best Paper Presentation Awards goes to the following presenters :

1. Himanshu Ranjan

Kashyap, Gauhati University

2. Dr. Bimal Kr. Kalita,

Assam Don Bosco University

3. Pallavi Saikia, Gauhati

University

4. Dr. Sandip Bordoloi, GCU,

5. Dr. Sanjib Kalita, Center of

Plasma Physics - Institute for Plasma Research

The Research Conclave 2024 successfully achieved its objective. It ends with the vote of thanks given by Prof. Minakshi Gogoi, HoD, Department of CSE, GCU, Assam



GIRIJANANDA CHOWDHURY UNIVERSITY

RESEARCH CONCLAVE 2024

GCU ASSAM

AI's Future: Merging Limits & Inspiring Innovation

28 NOVEMBER 2024

KEYNOTE SPEAKER



Dr. Shyamanta M Hazarika
Professor, Dept. of Mechanical Engineering and
Mehta Family School of Data Science &
Artificial Intelligence, IIT Guwahati

INVITED SPEAKER



Dr. Samarjeet Borah
Professor and Head, Department of Computer
Applications, Sikkim Manipal Institute of
Technology, Sikkim Manipal University, India

Organized by
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GIRIJANANDA CHOWDHURY UNIVERSITY, ASSAM

**AICTE-Vibrant Advocacy for Advancement and
Nurturing of Indian Languages**



GIRIJANANDA
CHOWDHURY
UNIVERSITY



**2 DAYS WORKSHOP ON
ARTIFICIAL INTELLIGENCE - RESEARCH
PERSPECTIVE IN MULTI-DISCIPLINARY FIELDS**

**“কৃত্রিম বুদ্ধিমত্তা-বহু-বিষয়ক ক্ষেত্রত
গৱেষণা দৃষ্টিভঙ্গী” - দুদিনীয়া কৰ্মশালা**

(Emerging Field: Advanced Computing, Supercomputing, AI, Quantum Computing)

Under AICTE-VAANI scheme in Assamese Language
এ আই সি টি ই-ভানী আঁচনিৰ অধীনত

29 - 30 November, 2024

২৯-৩০ নৱেম্বৰ, ২০২৪

Department of Computer Science & Engineering is going to organize 2 Days Workshop on Artificial Intelligence -Research Perspective in Multi-disciplinary fields sponsored by All India Council for Technical Education (AICTE), New Delhi under the Scheme of AICTE-Vibrant Advocacy for Advancement and Nurturing of Indian Languages from 29th to 30th November 2024.

The primary objective of the workshop, conducted in Assamese, is to convey the research perspective of artificial intelligence in various fields. This ensures that language does not hinder knowledge acquisition. Most of the resources are available in English and this workshop will help to bridge the gap.

From the technical point of view, the objectives are:

1. Explore the Future of AI
2. Examine AI Applications in Healthcare and Earth Science
3. Enhance Skills in Data Analytics
4. Discover Advanced AI Technologies



The Department of Computer Science and Engineering at Girijananda Chowdhury University organized a successful One-Day Hands-on Skill Training on 6th November 2024 for 24 students from Loharghat Higher Secondary School.



Empowering the Civilian Staff of Indian Air Force!

We're thrilled to share the success of our Corporate Training on Office Automation & Data Handling, conducted by Computer Centre, GCU!

21st Oct - 2nd Nov

40 hours of intensive training

Hands-on sessions on MS Office, Google Workspace & more

Data handling techniques for efficient management

Participants gained valuable insights, enhancing their skills & productivity!

Special thanks to our trainers: Rubi Kalita , Mridushmita Baruah, Gargee Chakrabarty, Subungshri Basumatary

Event Coordinator: Deepjyoti Deka

Certificate Distribution by Deputy Registrar & HOD, CSE Dept.

Boosting professional development & contributions!

Alumni Talk
September 1, 2024



Placement Preparation



SPEAKER

Mr. Subhranil Ghosh
Customer Service Associate
Federal Bank

Organized By
**Department of
Computer Science and Engineering**

Webinar

To know more visit
<https://gcuniversity.ac.in>

Alumni talk on “Placement Preparation” was organized by Dept of Computer Science and Engineering on 1st September 2024. Alumni, Mr. Subhranil Ghosh of 2019 batch who is currently working as Customer Service Associate in Federal Bank was invited to deliver the talk.



Guwahati, Assam, India
1, near SUBHAM REGENCY, Azara, Guwahati, Assam 781017, India

The Department of Computer Science & Engineering at Girijananda Chowdhury University, Assam, successfully organized a two-week Summer Internship Program on “Artificial Intelligence & Machine Learning” from July 22, 2024, to August 3, 2024.

INDUSTRY



An industrial visit was conducted on 11th April 2025 for the B.Tech 6th Semester students of the Computer Science & Engineering (CSE) department of Girijananda Chowdhury University, Assam to the Software Technology Parks of India (STPI), Borjhar, Azara. A total of 38 students participated in the visit,

accompanied by faculty members Ms. Shrabani Medhi, Ms. Mala Ahmed, Ms. Dharitri Sarkar, and Ms. Gitanjali Devi from the CSE Department.

Industrial Visit to AMTRON, Education & Training Center, Assam on 31st January 2025 for B.Tech 2nd year students of Computer Science and Engineering department.

Total 32 students of B.Tech (CSE) 4th semester along with two faculty members Ms. Shrabani Medhi and Ms. Mala Ahmed from the Department of Computer Science of Engineering went to AMTRON.



ACHIEVEMENTS

GCU RESEARCH EXCELS



Dr. Subungshri Basumatary
Assistant Professor, CSE, GCU



Anurag Bhuyan
7th Sem, CSE, GCU



Nasik Ahmed
7th Sem, CSE, GCU



Rohit Singha
7th Sem, CSE, GCU

GIRIJANANDA CHOWDHURY UNIVERSITY

Our students **Rohit Singha, Nasik Ahmed, Anurag Bhuyan** of CSE 7th Semester, GCU, under the guidance of **Dr. Subungshri Basumatary, Assistant Professor, CSE, GCU**, have achieved the best online presenter award on their research paper titled "A diabetes prediction model using various Machine Learning approach" at the 2nd International Conference on Data-Driven AI (ICDAA-2025), organized by The Assam Kaziranga University, held from 7-9, 2025.

GCU RESEARCH EXCELS



Ms. Shrabani Medhi
Assistant Professor, CSE, GCU



Sohini Roy
7th Sem, CSE, GCU



Sagarika Das
7th Sem, CSE, GCU



Sagar Choudhury
7th Sem, CSE, GCU

GIRIJANANDA CHOWDHURY UNIVERSITY

Our students **Sagar Choudhury, Sagarika Das, and Sohini Roy** of CSE 7th Semester, GCU, under the guidance of **Ms. Shrabani Medhi, Assistant Professor, CSE, GCU**, have successfully presented their research paper titled "A Web-Based Framework for Carbon Footprint Calculation and Visualization" at the 2nd International Conference on Data-Driven AI (ICDAA-2025), organized by Assam Kaziranga University, held from May 7-9, 2025.



An incredible 5-Day Workshop on Web Development with XT Academy at GCU !

A team of talented students from GCU showcased their innovation and technical skills at the hackathon held at Guwahati University on March 21, securing an impressive 3rd position.

Team Members:

Monish Bharali

Abhay Sharma

Arnab Kumar Das

Dibakar Bhattacharjee

Jit Roy



The team developed an AI-powered Financial Assistance App, designed to provide real-time stock market insights, investment guidance, and financial planning assistance. Their solution stood out for its intelligent analytics and user-friendly interface, earning them well-deserved recognition.

Congratulations to Mr. Nikhil Kumar Nayak from ECE 6th Semester and Mr. Abhay Kumar Sharma from CSE 6th Semester for securing the third prize in the COLLEGE QUIZ 2025, organized by The Hindu FIC & Institute of Public Enterprise (IPE).

Additionally, another team consisting of Mr. Debasish Baschyas and Mr. Bidip Talukdar from CSE 4th Semester showcased their excellence by qualifying for the Finals.



The competition witnessed the participation of 65 teams, making their achievement even more commendable.

Congratulations

to the winners of
SMART INDIA HACKATHON 2024



**TEAM TERABYTE - GIRIJANANDA
CHOWDHURY UNIVERSITY**

Problem Statement - Conversational Image Recognition Chatbot

Media Partner



Official Partner



Evaluation Partner



Platform Partner



Learning Partner



Nodal Center
Sponsor



GCU celebrated the achievement of Team Terabyte, winners of the 1st Prize at the Smart India Hackathon 2024, held at Welingkar Institute of Management, Mumbai, on 11–12 December 2024. The team secured a cash prize of ₹1 lakh and received a warm felicitation from the GCU community in the presence of the Hon'ble Vice Chancellor, Registrar, Deans, and HoDs.

Their mentor, Mr. Adarsh Pradhan, Assistant Professor, CSE Department, emphasized the value of perseverance and learning from failure. Team members shared their experiences and thanked the university for its support.

Recognizing their achievement, the Vice Chancellor announced an additional reward of ₹1.5 lakh for the team and their mentor, calling it a proud milestone for GCU and urging everyone to continue motivating students toward excellence.



Two teams from Girijananda Chowdhury University reached the finals of the Col. Guru Prasad Das Hackathon, organized by the JEC Alumni Society in collaboration with Jorhat Engineering College.

From the Department of Computer Science and Engineering, Ritanjit Das and Sumant Das represented GCU as part of the finalist teams under the mentorship of Mr. Adarsh Pradhan and Mr. Ajaan Anubhav Borah, Assistant Professors.

Adding to the pride, Group 1, which included Ritanjit Das, secured the 4th position in the competition.



Congratulations to Akash Bora and Rishav Kumar Singh from Girijananda Chowdhury University for winning **1st Prize** in "The Inspiration"—a 24-hour national-level hackathon held at IIIT Guwahati as part of E-Summit INIZIO 2025, in collaboration with STPI Guwahati!

Their project, Terapic Node, is a cutting-edge node-based photo editing platform that revolutionizes image transformation with advanced processing methods and an intuitive interface.



Girijananda Chowdhury University congratulates Prasurjya Goswami for securing the 1st Prize in the Cartoon Art Competition at EUPHISM '25, held in March 2025.

His creative expression and artistic excellence brought pride to the Department of Computer Science and Engineering and the entire GCU community. The university commends his achievement and encourages students to continue showcasing their talents in diverse creative platforms.



Girijananda Chowdhury University (GCU) proudly congratulates Ayon Chila Oja for bringing glory to the institution by winning the Bronze Medal in Mimicry at the 38th AIU Inter University National Youth Festival held at Amity University, Noida from 3rd–6th March 2025.

Competing among top talents from universities across India, Ayon showcased exceptional creativity and performance skills, making GCU proud on the national stage. His achievement stands as a testament to his dedication and the university's continued encouragement of artistic excellence.



PLACEMENTS



**GIRIJANANDA
CHOWDHURY
UNIVERSITY**

**School of
Engineering
and Technology**

Congratulations



**Uddipan Das
(CSE)**



**Ajita Koirala
(CSE)**

for getting selected in the virtual campus drive of SkillForge
E-Learning Solutions Private Limited.



**GIRIJANANDA
CHOWDHURY
UNIVERSITY**

**School of
Engineering
and Technology**

Congratulations



Himangshu Baro
Computer Science & Engineering

for getting selected in the campus drive of KnackTech
E-Learning Pvt. Limited.



**GIRIJANANDA
CHOWDHURY
UNIVERSITY**

**School of
Engineering
and Technology**

Congratulations



Dibashree Baruah
Computer Science & Engineering

for being placed in LTIMindtree



 GIRIJANANDA CHOWDHURY UNIVERSITY

CONGRATULATIONS
SCHOOL OF ENGINEERING & TECHNOLOGY



BAHNIMAAN KASHYAP
COMPUTER SCIENCE & ENGINEERING

FOR BEING PLACED IN TVS CREDIT SERVICES LTD.



 GIRIJANANDA CHOWDHURY UNIVERSITY

CONGRATULATIONS
SCHOOL OF ENGINEERING & TECHNOLOGY



UDDIPAN DAS
COMPUTER SCIENCE & ENGINEERING

FOR BEING PLACED IN SMART ED



 GIRIJANANDA CHOWDHURY UNIVERSITY | School of Engineering and Technology

Congratulations



Trisha Devchoudhury
Computer Science & Engineering

for being placed in Jyesta Corporate Entity Pvt. Limited.



 GIRIJANANDA CHOWDHURY UNIVERSITY | School of Engineering and Technology

Congratulations



Trisha Devchoudhury
Computer Science & Engineering

for getting selected in the virtual campus drive of Numetry



 GIRIJANANDA CHOWDHURY UNIVERSITY | School of Engineering and Technology

Congratulations



Himangshu Boro
Computer Science & Engineering

for being placed in Launched Global



CREATIVE CORNER



Ikhyt Dee Neag



Pratyush Bikash Choudhury



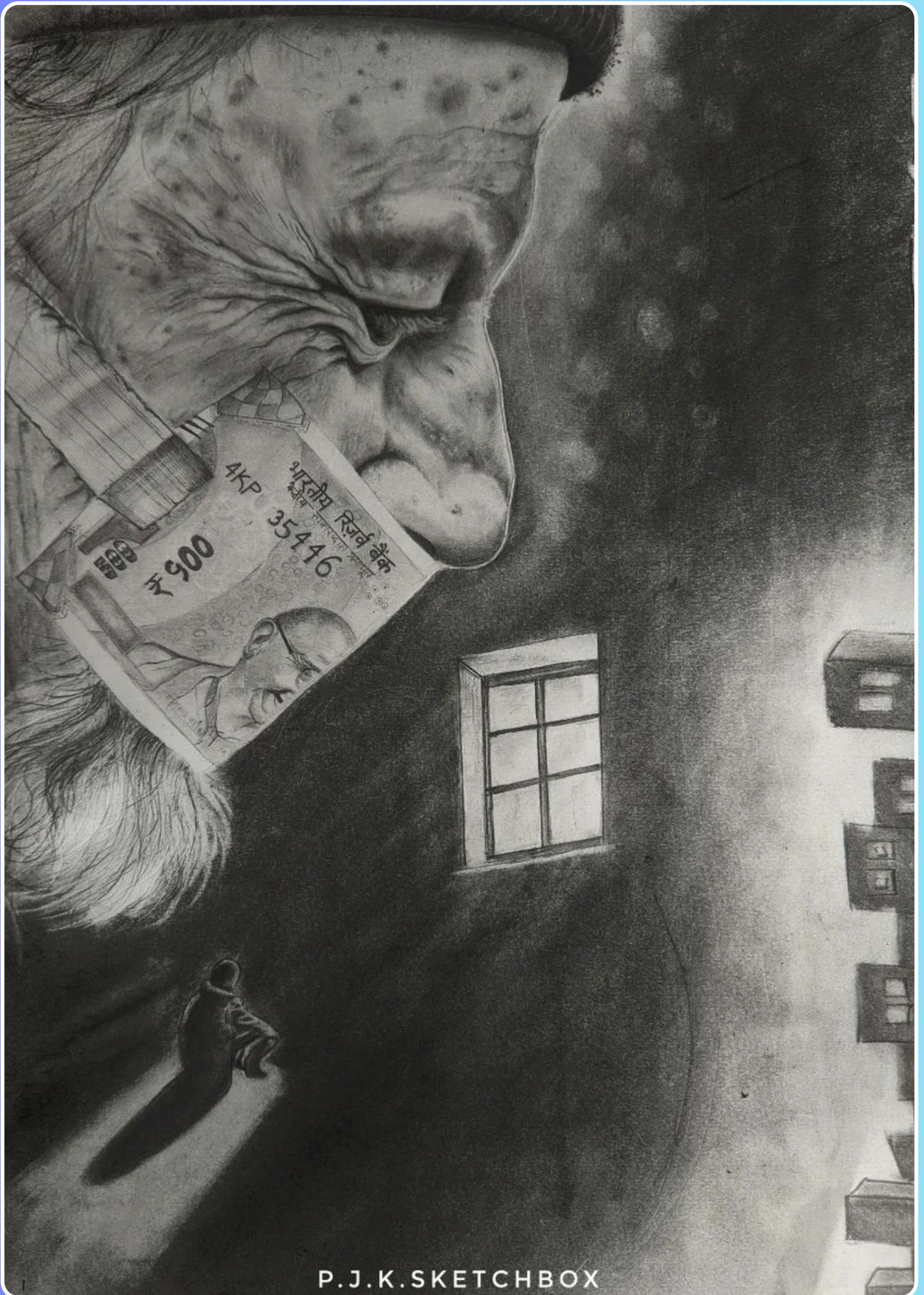
Bhargab Talukdar



Ashutosh Dutta



Harshita Nath



P.J.K.SKETCHBOX

Parag Jyoti Kalita

FACULTY & STAFF



Mr. Dipjyoti Deka, Ms. Manjula Kalita, Ms. Shrabani Medhi, Prof. Minakshi Gogoi, Dr. Th. Shanta Kumar, Mr. Adarsh Pradhan, Mr. Amarjyoti Pathak

(Sitting from Left to Right)

Ms. Gargee Chakravarty, Mrs. Rubi Kalita, Mrs. Gitanjali Devi, Mrs. Mridusmita Baruah, Ms. Mala Ahmed, Ms. Dharitri Sarkar, Ms. Ritushree Dutta, Dr. Monisha Devi, Subungshri Basumatary

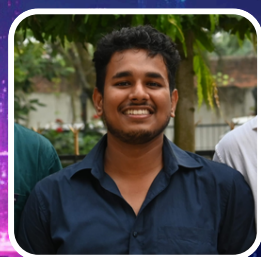
(Standing From Left To Right)

Absentee:

Mr. Arindam Saha, Mrs. Pinky Saikia Dutta



Cover Page Designer,
Ayon Chila Oja
Dept. of CSE, 4th Sem



Cover Page Designer,
Dwij Raj Bhagawati
Dept. of CSE, 2nd Sem