



HITech

CSE Departmental Magazine



*September
2024 Edition*

Message from the Mentor's Desk



Jhalak Dutta
Assistant Professor

Welcome to the new edition of our departmental magazine!

This publication is a vibrant blend of creativity and intellect, where emotions, societal issues, technical innovations, and achievements come together to inspire and connect us all. It stands as a testament to the creativity and dedication of our contributors, who have poured their passion into every piece, making each page a source of inspiration. I would like to extend my heartfelt thanks to all the students, faculty members, and alumni who have contributed to this edition. Your insights, creativity, and dedication have truly enriched this magazine, making it a vibrant platform for sharing knowledge and ideas. A special word of gratitude goes to our Head of the Department of Computer Science and Engineering, whose unwavering support and guidance have been instrumental in bringing this edition to fruition.

I also want to acknowledge the editorial team for their tireless efforts in curating and finalizing this edition with such dedication and precision. Finally, I would like to express my gratitude to our readers. Your engagement and support transform this magazine into more than just a collection of articles; it becomes a shared journey of discovery and inspiration.

Happy reading!

This is a very special edition of the CSE departmental magazine. A new breath of life has been infused in the magazine with the on-boarding of our wonderful student editorial team. And their enthusiasm has been reflected in the pages of the magazine. We extend a warm welcome to every member. We also expect that this would inspire an even more number of students to come forward with their contribution for the magazine. The magazine is now, in the truest sense, for the students and by the students. Please feel free to mail your articles on technological developments, product/software/book reviews, hackathon ideas and experiences, prose, poetry, and photographs to the magazine email address anytime, throughout the year. Being published in the departmental magazine is an opportunity that should not be missed!

Beyond the pages of this magazine, we hope you find inspiration in your surroundings and people to become the best version of yourself. We hope you give adequate attention to your physical and mental health and emerge as responsible and happy individuals.

Happy reading!



Reshma Roychoudhuri
Assistant Professor

Message from the Editorial Board

Welcome to this year's first edition of our departmental magazine!

We, the students of Computer Science and Engineering Department of Heritage Institute of Technology take immense pride in presenting Hi-Tech. We are thrilled to take this step in order to showcase diverse talents and achievements of our department. The process of creating this magazine has been a testament to the dedication and enthusiasm of our contributors.

The Editorial Team extends its heartfelt gratitude to our mentors, Professors, and our respected Head of the Department Prof. (Dr.) Subhashis Majumder.

We also express our gratitude to our mentors Prof. Jhalak Dutta and Prof. Reshma Roychoudhuri for their guidance and unwavering support.

Our magazine is a mosaic of voices, each offering a unique perspective on various aspects of our departmental life. You'll find insightful articles, thought provoking opinion pieces, and creative expressions that catch the essence of our department. This edition also serves as a platform to celebrate the numerous achievements of our students, faculty and staff.

As we look ahead, we are excited about the future possibilities for our magazine. We envision it growing into a platform that fosters dialogue, encourages collaboration, and promotes a sense of belonging within our community.

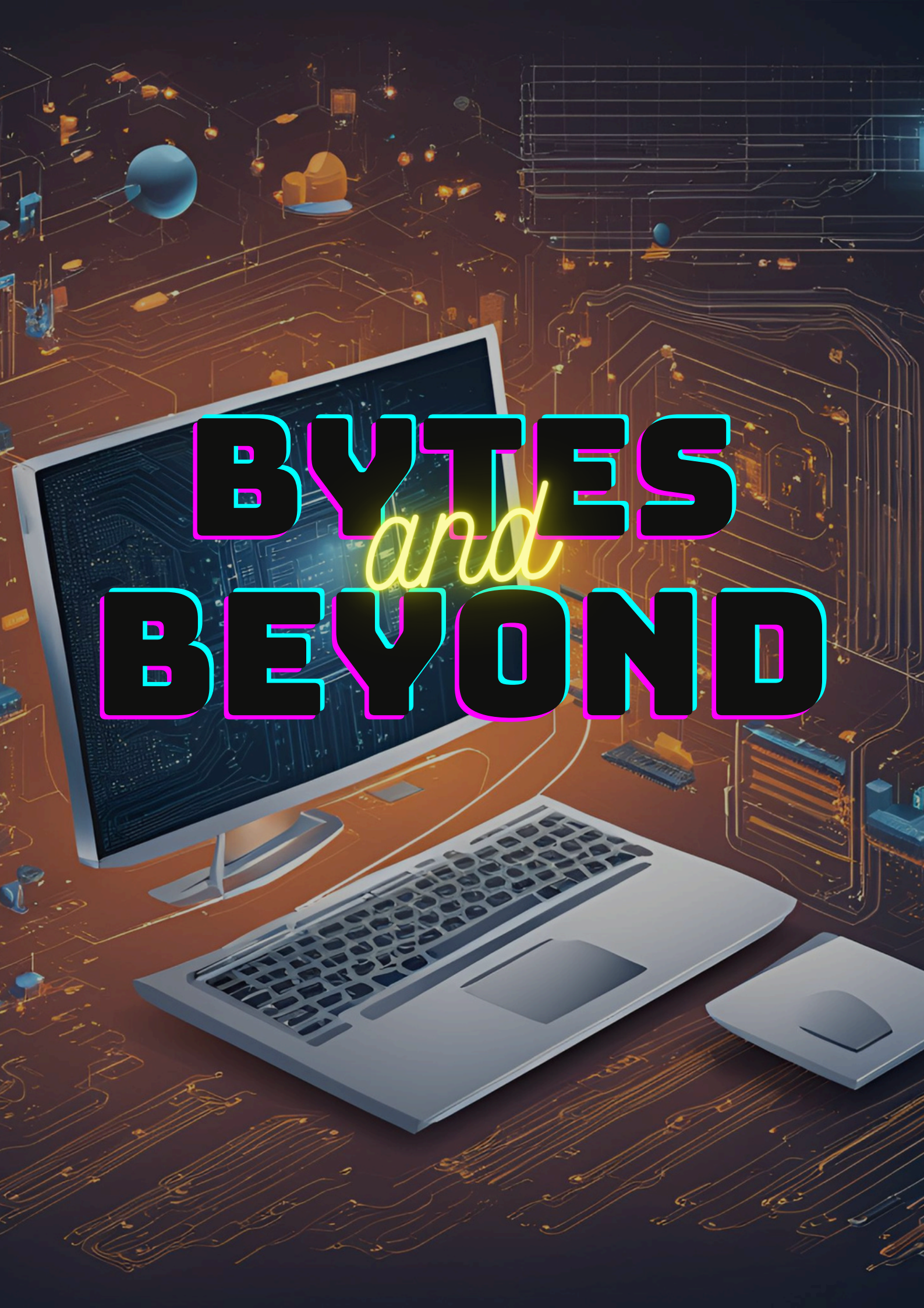
As you turn the pages, we invite you to reflect on the incredible journey we have embarked upon together and look forward to exciting adventures.

Warm regards,
The Editorial Board

TABLE OF CONTENTS

1	Bytes and Beyond	4
2	Pages of Perspective	19
3	Lens and Light	30
4	Word Weaver	35
5	Colour Conversations	47
6	Victory Vault	51
7	Legacy Spotlight	55
8	Student Innovation	64
9	Inside Scope	67



The background is a vibrant, futuristic digital landscape. It features a complex network of glowing orange and yellow circuit lines that crisscross the entire scene. In the foreground, a sleek, modern computer setup is visible, including a large monitor displaying a circuit pattern, a keyboard, and a mouse. The overall color palette is dominated by deep blues, oranges, and yellows, creating a high-tech, cybernetic atmosphere. The text "BYTES and BEYOND" is prominently displayed in the center, with "and" in a cursive script and the other words in a bold, blocky font with a glowing cyan outline.

BYTES *and* BEYOND

The Tale of ProgLand: The Great Parameter Passing Debate

-Debasish Jana
Professor

In the mystical world of ProgLand, where code flowed like rivers and algorithms sprouted like trees, an ancient debate raged on: the eternal struggle between "Pass-by-Value" and "Pass-by-Reference." Programmers across ProgLand knew that understanding these concepts was crucial to mastering the wisdom of experienced developers.

The Gathering of the Greats

In the heart of ProgLand, under the grand dome of CodeHall, four mighty communities—C, Python, Java, and C++—convened to settle the matter once and for all. Each community had its unique way of handling parameter passing, convinced that their method was the best.

C, the eldest of the languages, was a wizard with a long white beard that flowed like a stream of pointers. Python, a sleek and efficient sorceress, was known for her elegance and straightforwardness. Java, a thoughtful and methodical prince, often scratched his head in contemplation. And C++, a powerful knight, wielded the flexibility of both C and more advanced methods with confidence.

As the leaders took their places, a hush fell over the crowd. The debate was about to begin.

The C Way: The Myth of Value Practice

C, the wise old wizard, stepped forward, his staff tapping rhythmically on the floor. "In the land of C," he began, his voice carrying the weight of centuries, "we pass everything by value!"

The C community revered the "Value Practice," an ancient protocol followed with unwavering faith. Whenever a FunctionMaster needed to exchange messages—known as passing arguments—they always created a copy of the message first. "Why do we copy everything?" curious onlookers would ask. The Value Practice Leaders would smile and say, "To keep the original safe, of course! We wouldn't want a FunctionMaster to accidentally mess up the original message."

C raised his staff, and a shimmering copy of an integer appeared in the air. “If you wish to manipulate the original data,” he intoned, “you must wield the power of pointers!” He demonstrated with a spell:

```
void passValue(int data) {  
    // data gets the copy of original  
    data = 20; // local copy gets changed  
}  
void changeValue(int *ptr) {  
    // ptr gets the copy of original  
    // pointer  
    *ptr = 20; //content of ptr changed  
}
```

```
int main() {  
    int no = 10;  
    passValue(no); // number passed by  
    value  
    printf("%d\n", no); // Outputs 10  
    changeValue(&no); // pointer passed  
    by value  
    printf("%d\n", no); // Outputs 20  
    return 0;  
}
```

The crowd watched in awe as the integer changed. “But remember,” C warned, “there is no such thing as ‘call by address’ or ‘call by pointer.’ When you pass an address or a pointer, you’re still passing it by value—it’s the value of the pointer that’s being passed, not the actual memory location it points to.” To clarify, he continued:

```
void changePointer(int *ptr) {  
    int tmp = 30; // a local variable  
    ptr = &tmp; // ptr points to local  
}
```

```
int main() {  
    int no = 10;  
    int *pointer = &no;  
    changePointer(pointer); // pointer  
    unaltered  
    printf("%d\n", *pointer); // Outputs  
    10  
    return 0;  
}
```

The crowd murmured in understanding, realizing that pointers were simply another kind of value in C.

The Python Approach: Pass-by-Object-Reference

Next, Python, the elegant sorceress, stepped forward, her robes flowing like streams of code. “In my land,” she began, her voice as smooth as a well-written script, “we prioritize efficiency. Why waste time and resources copying messages when the FunctionMasters can handle the originals directly?”

The Python community followed a different practice. Whenever a message needed processing, the original message itself was handed over. Python waved her hand, and a scroll appeared, floating in mid-air. “In Python, everything is an object,” she explained, “and I pass references to these objects by value. This means you can modify the object’s content, but if you reassign the reference, the original remains untouched.”

She demonstrated:

<pre>def change_value(box): box['value'] = 20 def reassign_reference(box): box = {'value': 5}</pre>	<pre>box = {'value': 10} change_value(box) print(box['value']) # Outputs 20 reassign_reference(box) print(box['value']) # Still outputs 20</pre>
--	--

The crowd clapped and laughed, appreciating Python’s flexibility. “But what if the FunctionMasters accidentally changed the original message?” someone asked. Python smiled enigmatically, “That, my dear, is the risk we take for efficiency.”

Java’s Confusion: Pass-by-Value with a Twist

Java, the thoughtful prince, stepped forward next, a slight frown of concentration on his face. “In my land,” he began, “we are always taught that we pass by value, but there’s a twist. When we pass primitive types like int or boolean,” Java explained, “we make a copy, just like in C. But when we pass objects, we pass the reference to the object by value. It’s like handing over the address where the original message is stored, not the message itself.”

The Java developers in the crowd scratched their heads. “So, are we passing by value or by reference?” they asked. “No, no,” Java replied, trying to clarify. “You’re passing the reference by value. It’s still pass-by-value, just for the reference itself!”

He continued:

<pre>class MagicBox { int value = 10; } public class Main { public static void changeValue[MagicBox box] { box.value = 20; } }</pre>	<pre>public static void main[String[] args] { MagicBox box = new MagicBox[]; changeValue[box]; System.out.println[box.value]; //Output s 20 } }</pre>
--	--

The crowd was intrigued, though some were still puzzled. "So, it's not pass-by-reference?" they asked again. "No," Java insisted, "it's pass-by-value... of the reference!"

The Pragmatic C++: The Best of Both Worlds

Finally, C++, the powerful knight, took the stage. "In my land," he said with a confident grin, "we have the best of both worlds. Why choose one when you can have both?"

In C++, developers had the choice to pass by value or by reference, depending on what was needed. When they wanted to keep things safe and simple, they used value practice like in C. But when they wanted efficiency, they used references. C++ demonstrated his process:

<pre>void changeValue[int &ref] { // data not local but alias of original ref = 20; // original gets modified } void changePointer[int *&ptr] { int tmp = 30; // a local variable ptr = &tmp; // ptr points to local }</pre>	<pre>int main[] { int no = 10; changeValue[no]; cout << no << endl; // Outputs 20 int *pointer = &no; changePointer[pointer]; // pointer altered printf["%d\n", *pointer]; // pointer dangling return 0; }</pre>
--	--

As the crowd cheered, a scholar raised a hand. "But Sir C++, what about passing pointers? Can you pass them by value or by reference?" C++ nodded, pleased with the question. "Ah, indeed! In C++, you have the flexibility to pass pointers either by value or by reference, unlike in C where you can only pass them by value."

He added with a grin, "But beware! In C++, a reference variable should never refer to a pointer that points to a local variable. When the function returns and control goes out of scope, the local variable is destroyed, leaving the pointer dangling and pointing to a memory location that no longer holds valid data."

A Witty Interlude

As the discussions continued, a distinguished figure, Professor ProgWizard, emerged from the shadows. With a twinkle in his eye, he addressed the crowd. "Ah, parameter passing, an age-old debate indeed. Let me share a witty remark by Niklaus Wirth, the creator of the Pascal language. His name, pronounced like 'Virt' in Europe, is often pronounced 'Worth' in the US. He once quipped, 'In Europe, I'm called by name, but in the US, I'm called by value.'"

The crowd chuckled, appreciating the clever pun. It lightened the atmosphere and reminded everyone that even the most serious debates could have a touch of humor.

The Grand Revelation

As the sun set over ProgLand, the leaders of the communities gathered to reflect. They realized that each language had its own strengths and peculiarities when it came to parameter passing. There was no one-size-fits-all answer.

Professor ProgWizard, now at the center of the gathering, spoke up, "Dear citizens of ProgLand, let me clarify once and for all. In C, you always pass by value, even when using pointers. Pointers themselves are passed by value, though you can modify the data they point to. In Java, you pass references by value, which confuses many into thinking it's pass-by-reference. In Python, it's pass-by-object-reference—an elegant approach that lets you modify objects but not reassign references. And in C++, you have the power to choose between passing by value and by reference, including pointers."

He paused, letting the weight of his words sink in. "Remember, there is no such thing as 'call by address' or 'call by pointer.' If an address or pointer is passed, it is passed by value, except in C++ where you can pass a reference to a pointer."

With that, the leaders of C, Python, Java, and C++ nodded in agreement. The great debate of ProgLand was settled, not by deciding which method was best, but by understanding the nuances of each.

The Conclusion

The citizens of ProgLand returned to their code, wiser and more united in their understanding. They knew that mastering the art of parameter passing was crucial, but they also knew that each language had its own way of doing things. And that was perfectly fine.

The dome of CodeHall echoed with newfound wisdom as the sun dipped below the horizon, signaling the end of a fruitful day in the land where code was king.



Doomscrolling- The bane of the Internet age

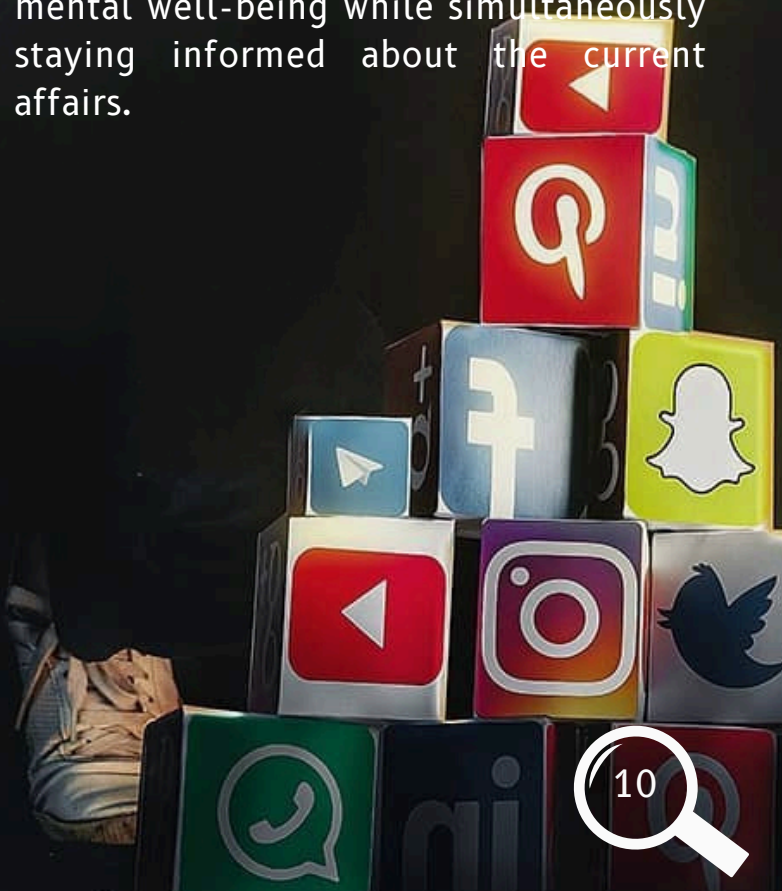
-Yubaraj Das
3rd Year

Have you ever found yourself lying in the bed at night, constantly scrolling through social media and news sites even though you know it's not good for you? You might even ask yourself, "Why am I doing this?". This behaviour is called doomscrolling. It refers to the habit of endlessly consuming news online, particularly negative news, on the internet and social media.



Doomscrolling is fuelled by the human brain's natural inclination towards negativity bias, where we are likely to focus more on and remember the negative experience over the positive ones. This is the reason why despite the emotional toll, many find themselves unable to break the cycle, endlessly scrolling through a stream of bad news. The impact of doomscrolling on mental health cannot be understated. Prolonged exposure to negative content can lead to feelings of helplessness, anxiety and depression.

"Doomscrollers" often sacrifice sleep, leading to disrupted sleep patterns and decrease in overall well-being. Meanwhile, the algorithms of the addictive social media platforms are designed to keep users engaged, thus amplifying the consequences. Establishing a healthy relationship with technology requires conscious effort. The best way to tackle this issue is to set time limits on social media use and curate our news feed to more positive or neutral content. Meditating daily and performing a "social media detox" once every week will be beneficial too. In conclusion, while staying informed is important, it is equally crucial to recognize the detrimental effects of doomscrolling. By taking proactive steps to balance our social media consumption, we can protect our mental well-being while simultaneously staying informed about the current affairs.



Quantum Computing

-Aniruddha Shit
3rd Year

Understanding Quantum Computing

Quantum computing leverages quantum mechanics to perform computations beyond classical computers. Traditional computers use bits [0s and 1s], while quantum computers use quantum bits, or qubits, which can exist in multiple states simultaneously due to superposition.

The Power of Superposition and Entanglement

Another key principle is entanglement. When qubits entangle, the state of one qubit instantly influences another, regardless of distance. This allows quantum computers to solve complex problems quickly.



A handout picture shows Sundar Pichai and Daniel Sank with a Google quantum computer at the company's campus in Santa Barbara, California, in October 2019.

Potential Applications of Quantum Computing

Quantum computing has vast potential across fields. In cryptography, it could break current encryption, prompting quantum-resistant encryption. In medicine, quantum simulations could lead to drug discoveries by modeling molecular interactions. Optimization problems in logistics and finance could be solved more efficiently, saving time and resources.

Current Challenges and Future Prospects

Despite its promise, quantum computing is in its infancy. Researchers are working on overcoming technical challenges like qubit stability and error correction. As these are addressed, quantum computing will likely transform industries and redefine technology.

Linear and Logistic Regression

- Sagnik Pramanik
3rd Year

Linear regression for beginners!

Linear regression is also called the “Hello World” of Machine Learning. I will demonstrate this using a simple example, something that everyone has come across in their academic life, which is the experiment.

“L vs T^2 for a simple pendulum”

The main thing to note is that the graph of L vs T^2 for a pendulum is a straight line. You conduct the experiment, plot the points, and try to draw a straight line with a ruler.

Unfortunately, you see that you are unable to join all the points in a straight line. You ask your teacher about it.

The teacher says :

“Just draw the best-fit line”

Notice how the teacher says “Best-fit Line”. This is EXACTLY what linear regression is!

In linear regression, you are given values x_i and y_i and you have to predict $\hat{y}$, given $\hat{x}$

What linear regression algorithm does is that it finds out the slope and y-intercept for the best-fit line, which is also called the weight and bias respectively.

The equation becomes $\hat{y} = w \cdot \hat{x} + b$,
[which is the same as the equation of any straight line, $y=mx+c$] We get the actual value of w and b by training the model using the data.

Linear regression is a powerful tool for modelling linear relationships and making predictions. It's easy to use and interpret, making it a great starting point for beginners in machine learning.

Logistic regression

After linear regression, the second in line is **logistic regression**.

First off, here is a short note: logistic regression is not like other regressions [linear regression]. Here, regression is a misnomer. It is a classification algorithm.

Now, before getting into the details of *simple logistic regression* [aka logit regression], Let's look at the basic idea.

—> It is a binary classification algorithm.

“Binary” implies that the output can be of two types.

“Classification” means that after giving input, the model will provide the output whether it is this or that of type1 or type2

—> You cannot have both outputs at the same time, It's either type 1 or type 2, cannot be both.

Now,

Let's analyse the equation

$$y = 1 / [1 + e^{-x}]$$

This is called the sigmoid function.

This function, also known as the logistic function, takes any real number 'x' as input and outputs a value between 0 and 1.

- $x < 0$: The function outputs a value closer to 0.
- $x \geq 0$: The function outputs a value closer to 1. [taking 0.5 condition along with class 1]

During model training, the model learns a set of weights and biases that best represent the relationship between the input features and the target variable. These weights are adjusted iteratively using an optimization algorithm, aiming to minimise the difference between predicted $\hat{y}$ and actual y

The sigmoid function's output can be interpreted as the probability of the instance belonging to the positive class. For example, if the output is 0.8, it means that there is an 80% chance that the given input data corresponds to class 1. So, the output class is "class 1"

Similarly, other examples:

Output Value | Output Class

-----+-----

0.1 | Class 0

0.4985 | Class 0

0.5. | Class 1

0.5001 | Class 1

0.9989 | Class 1

This is how simple/ binary logistic regression works. It's easy to use and interpret, making it a great algorithm for binary classification. It is commonly used in

- Medical Diagnosis: Based on the data, does the patient have the disease [class 1] or not [class 0]
- Credit Scoring: Based on credit score and other data, a person is risky [class 1] or safe [class 0]
- Email Spam Detection: Does an email belong to class spam [class 1] or class non-spam [class 0]

And many more.

“The Concurrency Revolution”

-Anurag Dutta
4th year

Sitting in your chair and waiting for your massive code to execute is tedious, isn't it? Ever thought of slashing that time? Then hey!! You are reading the perfect article today.

This is a strategy I am implementing at my office on a project. Let's dive right in. Concurrency is the cutting-edge technique of running multiple instruction sequences simultaneously. With advancements in technology such as multi-core processors, parallel processing, and multiple processes/threads can be executed concurrently. It leads to enhanced performance, better average response time, and optimal resource utilization. Making your job a bit smoother, isn't it? But every benefit comes with a price.

While leveraging concurrency, one should keep following in mind:

- 1.The scheduling policies of the operating system.
- 2.Thread overheads- which include context switching and fair memory management.

Any programming language that supports multithreading enables concurrency.

A few pitfalls along with the perks of concurrency are deadlock, starvation, race conditions, blocking, and non-atomicity.

The Ethics of A.I.

-Shreya Denre
2nd Year

The ethics of AI: How it matters and How to address it

As we stand on the brink of a new era, the relationship between AI and human cognition is gradually unfurling, promising advancements in various fields including healthcare, education, and technology. These advancements bring significant ethical implications as with great power comes great responsibility.

The pursuit of ethical AI is paradoxical in a world where ethics often takes a backseat. We expect our creations to have a moral compass, but we, as their creators, often do not.

Definition of Ethical AI:

Ethical AI means developing and deploying AI systems that prioritize fairness, transparency, accountability, and privacy, ensuring that technological innovation respects human values. Balancing innovation with responsibility is crucial for building trust and achieving sustainable progress.

Concerns and its Solutions:

1.Bias: AI systems learn from data which may contain inherent biases, leading to discrimination. Example, an AI algorithm used in hiring might favour candidates from certain demographics over others if trained on biased data. So, auditing AI

systems for biases and implementing diverse data, fairness-aware machine learning is required. Ensuring fairness promotes social equality and enhances AI reliability and acceptance.

2.Privacy:The extensive data required for AI can include sensitive personal information. Robust data protection measures like anonymization, differential privacy, and federated learning are essential to safeguard personal information.

3.Transparency: Many AI models, specially deep learning algorithms, function as “black boxes,” obscuring their decision-making processes. Developing techniques like Local Interpretable Model-agnostic Explanations [LIME] can improve transparency. Clear lines of responsibility must be established, to determine accountability for AI actions.

In **conclusion**, the pursuit of ethical AI is essential for the responsible progress of technology. By prioritizing fairness, transparency, privacy, and accountability, we can create AI that respects human values and drives innovation. As AI grows, collaboration between technologists, ethicists, policymakers, and the public is crucial to ensure that AI serves the common good and benefits the society as a whole.



Heads or Tails? Counting Unique Objects with Coin Toss Magic

-Himadri Chatterjee
3rd Year

Imagine, you start your own website and you want to keep track of some very useful analytics like: How many unique visitors visit your site in one day? A very naive approach is to store a list with their IP Addresses and increase the count whenever a new IP Address gets appended to the list. Things get worse if your site goes viral and the count may reach to millions!



In a very recent paper, Professor Saurav Chakraborty of Indian Statistical Institute, Professor Vinodchandran Variyam of University of Nebraska-Lincoln, Professor Kuldeep Meel of University of Toronto, discuss a very efficient way to approximate the number of distinct entries in a long list, remembering a small number of entries.

To understand their approach [the CVM Algorithm], let's say, you're reading the book, Hamlet and you want to find out how many unique words are there, but your working memory, a whiteboard has room for only 500 words. So, first you write the first 500 words skipping any repeat. When the space is full, you now stop reading and start to flip an unbiased coin for each of the words in your memory, i.e., the whiteboard. Heads, the word stays and Tails, you delete it. Now you're left with about 250 words.

Next, you come to the Round 1 of the algorithm. Keep going through Hamlet and when you come to a word that is already there in your list, you flip a coin. If it's a tail, you delete the word and Heads, the word stays in your list. You continue in this manner until your memory has 500 words. Then randomly delete the half, again based on the outcomes of 500 tosses.

Moving to Round 2, you follow the same process as Round 1 but this time we'll make it harder to keep a word.

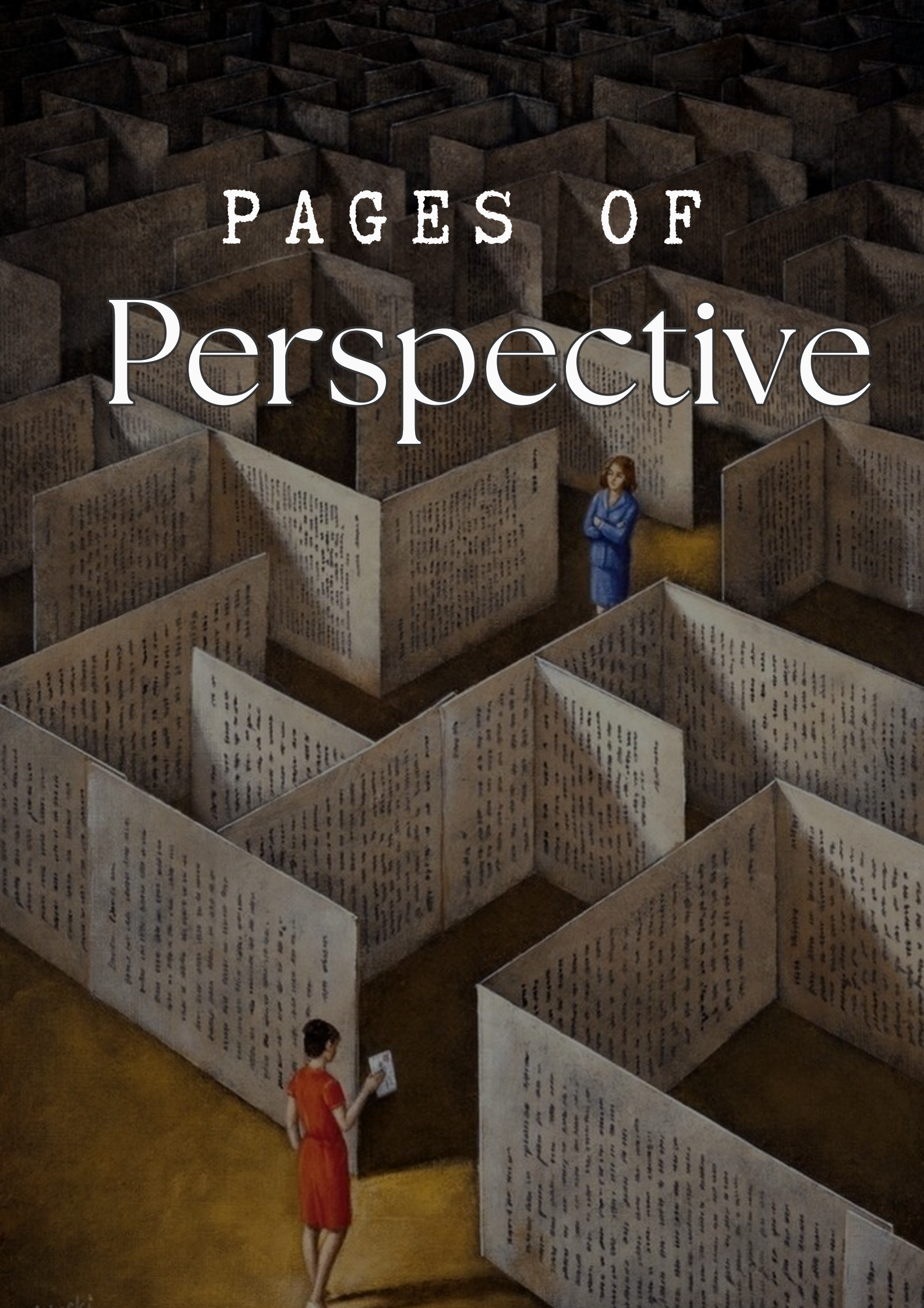
A repeated word will be deleted if it's a tail but will be kept only if you get 2 consecutive heads. At the end of Round 2 you'll again delete about 250 words, based on 500 tosses. Similarly, in Round 3, you'll need 3 consecutive heads to keep a repeated word and thus in kth round you need k consecutive heads and by virtue of the random selections every word has the same probability of being there: $1/2^k$.

Eventually, after nth round you come to the end of Hamlet. Suppose it took 4 rounds and after the final round your list has 246 words. Now it's a simple math to divide the final size of the list by the probability $1/2^4$ [since, it took 4 rounds and a fair coin has $\frac{1}{2}$ probability of getting each outcome]. The result of this above calculation comes out to be 3936. And they counted that Hamlet has exactly 3967 unique words. Now if the dataset is larger, with a bigger memory space, the efficiency of the CVM Algorithm might reach to 100%.

You may find the original work here:
<https://doi.org/10.4230/LIPIcs.ESA.2022.34>



PAGES OF Perspective



Time Management

-Sidhartha Mojumdar
Technical Assistant

Time spent on specific activities with conscious control over planning and execution is known as time management. This is done to maximize the output with an increase in effectiveness, efficiency and productivity. Time management is a challenge. A person has to know exactly what they are doing and what more needs to be done within a stipulated time frame.

The idea of time management has been practiced for 100 years, but time is almost uncontrollable. It was later found that it is more effective to control ourselves and use our time wisely.

The best way to manage our time is by beating time, achieving workload effectively, and repeating the same successfully. Sometimes it seems that we have never had enough time for work, though we all possess 24 hours a day. It is also sometimes observed that some people manage their time very cleverly. This is called time management.

There are several obvious benefits of time management:

- Increased productivity and efficiency.
- Reputation will be higher.
- Stress will be less than usual
- For advancement, opportunities will be increased.
- Important life and career goals can be achieved.

If someone fails to manage their time effectively, the results can be detrimental:

- Deadline missing
- Workflow inefficiency
- Degraded quality of work
- Reputation of poor work level
- Much higher stress level

But, if we follow some time management skills then everything will be much easier to achieve with huge benefits.

Tools for Time Management

To manage time efficiently, we need to have a strict routine so that every work is over by the deadline. Some tricks are as follows:

Activity Logs

By properly analyzing how we utilize our valuable time in the office, we can minimize or eliminate low-value activities. This will allow us to focus on higher-value tasks, resulting in more positive outcomes within the stipulated time frame.

An activity log, or activity diary can be made to write all our priority work ranging from higher priorities to lower ones. It also records our daily activities.

day. Our memory often poorly records activities, and relying solely on memory is likely to yield poor results. For example, we can say that if we do high priority jobs in the morning with a fresh state of mind, then there is a high chance that it will be done without any hackle. Low energy tasks like responding to the mail, or some returning calls can be done in the afternoon.

There are some particular ways to keep activity logs, such as if we have to open an Excel or spreadsheet and make some columns that are Important like Date/Time, activity description, remarks, time periods, value of tasks. Then, after achieving or completing all the job, we have to just note it down in some way so that it can



Activity logs can help us to have some knowledge about how we work during the right time of

be understood easily later. Negative habits that affect time management, like inattentive email replies or gossiping, do not contribute positively to future outcomes.

To do lists

Sometimes we forget to do high-priority tasks and waste valuable time on low-priority ones. All these are due to not maintaining a “to-do list”. To write this list, jot down the most prioritized tasks first, followed by the less important ones. Having this list ensures we have a clear index of work. To beat overload, a to-do list is the most important method. The key is to follow it accurately and strictly.



There are also some steps to follow in making lists.

First, we have to list down all the tasks and break them into small parts. Ideally, all tasks have to be completed within a stipulated time frame. Try to follow the pattern of works that are prioritized from A to F. A situation can happen when there is more than one high priority work. In that case, we have to break the list and have to manage the priority of jobs in between them. After this, we have to maintain this list as far as we are concerned .

Various software tools can help with maintaining a to-do list. These software tools can generate graphs to help us analyze our tasks more clearly. Lot of time management software exists such as MS Word, MS Excel. Gmail and Outlook express also help us to make the to-do lists in proper order.

Action Program

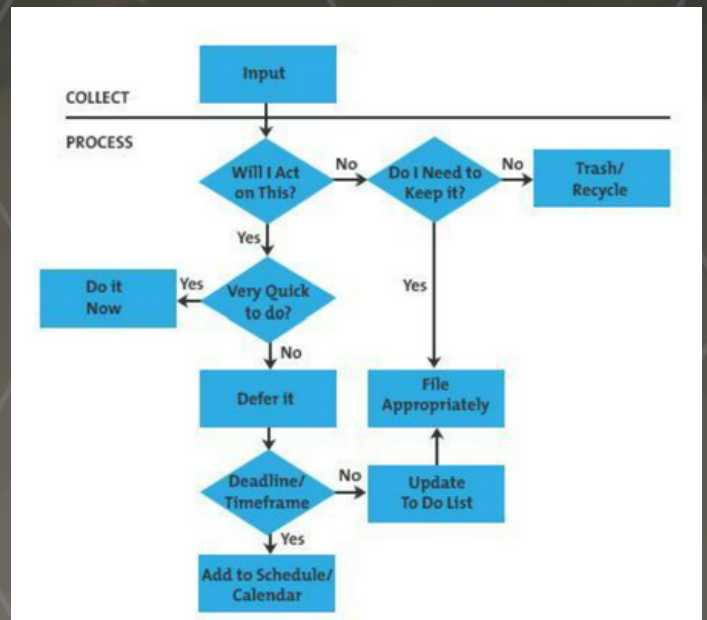
Becoming exceptionally well organized makes action programs, as we all know that to-do lists are important and helpful if we have a single job. But if we have multiple jobs in our hands, then it will become problematic to

manage only by some lists. This is a very common problem faced by managers because they actually handle a lot of projects at the same time. Some loopholes exist in to-do lists as they are not planned, focused or action-oriented. If the list becomes more than 20, then it becomes more tough and cumbersome to get. There is a method called an action program that is also called an industrial program. The main agenda is to collect all pending tasks and solve them systematically. Pruning is a step that involves assessing how much work has been done and what remains. There may be some jobs that are not relevant to us, but we have them on the list. In this case, we can just delete those parts. Organizing and prioritizing tasks are key to effective job scheduling.

Allen's Input Processing Technique

This is a process which manages workflow efficiency. Many of us have lots of information every day through emails, telephone, meetings, invoices etc. We have to be more careful to choose

important information from the entire media we have. So we must be more cautious about how we utilize this information.



- David Allen, a productive expert, developed the input processing technique, and he also has a book on this topic which was published in 2003. Allen defines it, "anything you have allowed into your psychological or physical world that doesn't belong where it is, but for which you haven't determined the desired outcome and step by step."
- This input can be anything that we have from sources like from emails, telephones etc. Benefits of this tool are that, by helping it in the way

- to take actions of all the inputs that are coming. There are three main stages that are helpful for the time management system:

1. Collecting
2. Processing
3. Reviewing

Valuing time

Time is equivalent to money, once time has passed, we cannot get it back. We must assess the value of our time.. This process should be followed rigorously so that we have to find tasks according to the client, which will give us more returns during a span of time. This method is a quick calculator to find the most lucrative work which will have some return.



Multitasking

Multitasking is the best quality of person to work the entire task in a good manner. For example,

a manager might take notes, answer calls, and respond to clients all while preparing an Excel file. All these tasks need to be done properly, without any errors.

This type of sequence of work is called multitasking. In a competitive environment, efficiency in multitasking is crucial for success. However, the quality of work often worsens when multitasking, but it may still be necessary to complete tasks on time.



Leverage

"Give me a lever long enough and a place to stand, and I can move the Earth."– Archimedes

To succeed in life we have to use leverage. Without it, we can have a certain degree of results. There are certain ways to apply leverage to our career, and certain ways to achieve it. For it, we have to know how to use the leverage of time, resources, knowledge and education with technology.



Time management is the main factor we can discuss here as far as practicable.

Conclusion

Time management is one of the most valuable qualities a person can possess. If we can use our time in a very positive way, then we can almost do the entire task in a stipulated time frame without any problem. But as we are human we cannot remember all the things at the same time, so for these we have to make some plans to achieve success at intervals of time. This study has explored various methods to increase productivity by applying strict rules.

References:

Kent. [2016]. *HOW TO MANAGE YOUR TIME EFFECTIVELY*. Retrieved from <http://www.kent.ac.uk/careers/sk/time.htm>

Mindtools. [2016]. *SMART Goals*. Retrieved from <https://www.mindtools.com/>: <https://www.mindtools.com/>

Skills you need. [2016]. *Time Management Skills*. Retrieved from <http://www.skillsyouneed.com/ps/time-management.html>

Time-management-guide. [2016]. *Personal Time Management Guide*. Retrieved from <http://www.time-management-guide.com/>

The most fundamental strategy is to use time in the most useful manner. We can work for many hours in a day, if we can manage the schedule by leverage. If we can use our time to increase the efficiency of our productivity, then it will be more helpful to achieve desired success.

What it means to write

-Avi Ambastha
3rd year

To write is to set yourself free. To write is to be liberated, like a flowing river. Just like rivers flow into the sea, the writer's thoughts flow into his pen.

He writes for himself, for the readers, for his loved ones, and for the ones he dislikes all alike. He creates worlds, stories, people, beliefs in them, and the relations or bonds they share. He has the power to make a piece of paper live in the reader's heart.

To write is like magic, isn't it?. To set your thoughts free on a piece of paper, to see your characters live through the minds of the readers, to see someone be inspired by your work; it's amazing.

Having the ability to write and to converse with one's thoughts through a piece of paper is magical. We study all sorts of sciences for human advancement, but we write for passion. Literature exists because our hearts burn with passion and love for it, because the human heart craves emotion and not just one, but every emotion—joy, sadness, victory, defeat, giving up on something, or never giving up. Every emotion that cannot be described by a single word finds a way to be expressed through characters in their journeys.

To write freely means to live freely, to burn with passion and love for this world. To see the world for what it is and to accept it. Every piece written is an art, and every reader is an admirer or a critic. Do not fear writing your heart out. Write—not for the praise of the readers, but to free your heart and let your thoughts flow. Write naturally, like a naturally flowing river, not like man-made fountains. Write with depths of the seas, with the grandeur of the mountains, and with love for all living beings. There is a great writer in every reader!!!

Juvenile Delinquency

-Devayanee Gupta
2nd Year

Juvenile delinquency, where minors engage in illegal activities, is a widespread issue affecting all socio-economic backgrounds. Both impoverished and affluent youths can become involved in criminal behaviour due to various factors. Understanding these diverse backgrounds is essential to address the root causes and implement effective solutions.

Children from poor families often face economic hardships, high stress levels, and inadequate supervision, leading them toward delinquency as a coping mechanism or to obtain material goods. For instance, parents working multiple jobs may leave children vulnerable to negative influences, resulting in behaviours like theft or drug trafficking.

Conversely, children from wealthy families might experience emotional neglect due to their parents' demanding careers or social obligations. These youths may engage in delinquency out of boredom or entitlement, participating in activities like underage drinking, drug use or vandalism.

Social media and influential figures significantly impact juvenile behaviour. Platforms like Instagram, TikTok, and YouTube often glamorize risky behaviours and criminal activities, making them appear desirable to impressionable youths. Influencers who engage in or promote such behaviours can encourage their young followers to emulate them. This normalization through social media creates a widespread

issue where kids are influenced to participate in illegal activities for fame, acceptance, or thrill-seeking.



Addressing juvenile delinquency requires a multifaceted approach. Strengthening family bonds is crucial, whether through economic support and accessible childcare for poor families or better work-life balance for wealthy ones. Providing educational and recreational opportunities, such as after-school programs and community-based mentorship, can positively engage youths. Additionally, regulating social media content to prevent the glamorization of criminal behaviour and holding

influencers accountable is essential. The legal system must balance accountability with rehabilitation, offering programs focused on education, vocational training, and counselling to prevent recidivism. Comprehensive efforts can mitigate juvenile delinquency and foster safer, more cohesive communities.

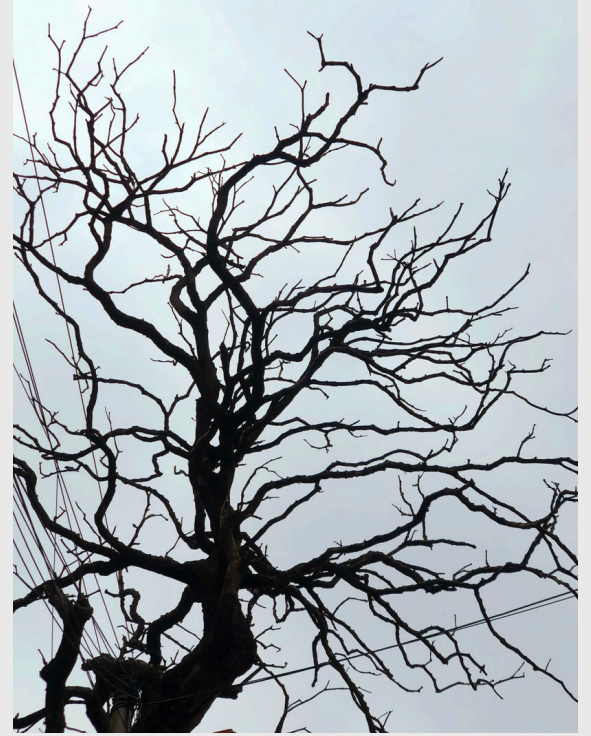




LENS *and* **LIGHT**

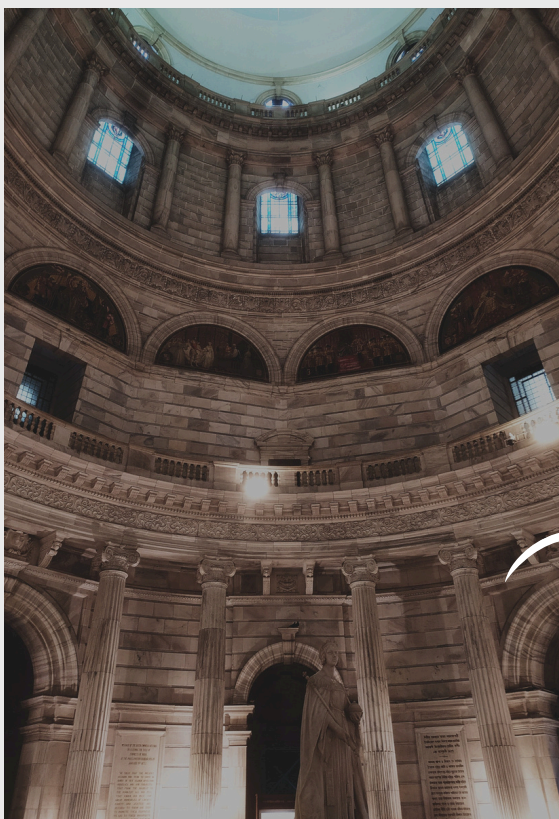
Ashutosh Bajpayee
3rd Year

Arnab Senapati
3rd Year



Anushka Sinha Ray

Anushka Sinha Ray
4th Year



Ankita Das
4th Year

Anuska Roy
2nd Year



Rohit Mondal
2nd Year



Anurag Sarkar
3rd Year

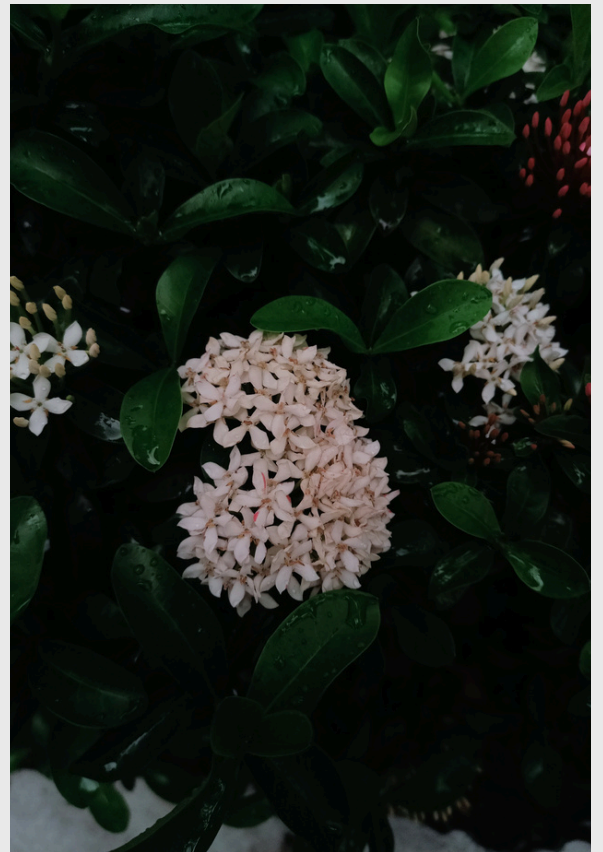




Ankita Sarkar
2nd Year



Sneha Agarwal
2nd Year



Aniruddha Shit
3rd Year

Sayantika Bhunia
3rd Year



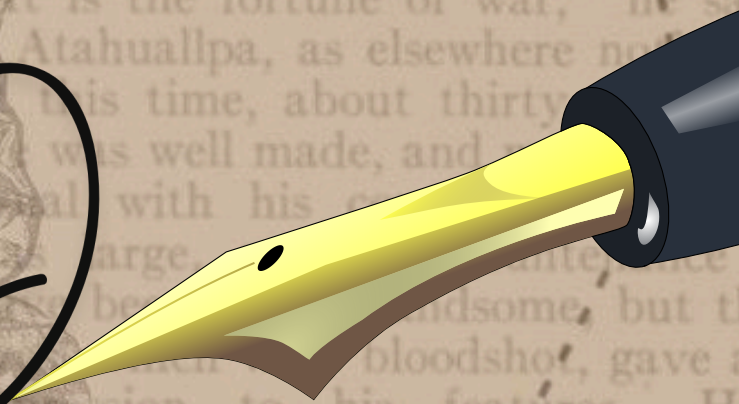
Saheli Rana
3rd Year



Ekam Bitt
3rd Year



WORD WEAVER



THE MIGHTY SEA

It was a time of joy,
as the vacation had just started.
On reaching her travel destination,
Divya was very excited.

It was for the first time
that the five year old was seeing the vast sea.
As its waves touched her tiny feet,
she started dancing with glee.

Divya said, “O mighty sea, O mighty sea,
Seeing you my happiness knows no bounds.
But, I wonder what makes you the happiest,
whenever you find it around.”

“Is it the sun that shines above,
giving your waters a golden hue?
Or is it the winds, soft and fresh,
that gently blows over you?”

Smiling the sea replied, “Little one,
I welcome you today.
But your guesses are wrong, my dear friend,
I am very sorry to say”.

“The sun definitely shines over me,
And the winds make my water swell.
But they don’t make me the happiest,
That I can surely tell.”

Thinking again, little Divya said,
“Then is it the ships big and small?
They sail on your waters day and night,
Spreading joy all along.”

“These vibrant ships sail on me,
And indeed I hold them dear.
But they don’t make me the happiest,
Your guess is wrong, I fear”

Unable to wait any longer,
the little girl said to the sea,
“Then what is it that makes you the happiest?
Please, tell that to me.”

“I feel the happiest when my waters touch
the tiny feet of innocent children.
Hearing their laughter, seeing their joy,
I am totally smitten.”

“Keep coming back to my shore, Divya
I will wait for you, my friend.
My waves will continue to play with you,
Till the day comes to an end.

-Dr. Anurina Tarafdar
Assistant Professor

GOD IS DEAD

What will you do tomorrow,
if you learn that God is Dead?
Will your purpose in life fade away?
or will you be consumed by dread?

Will you uphold what's right and bold?
in a world where justice remains untold,
or will you yield to your inner demon's cries,
letting your spirit shatter where darkness lies?

After you realize,
there is no one to save you,
Will you embrace your responsibility,
and finally see your actions through?

After you realize,
there's no hand to guide,
will you succumb to deceivers,
letting them control your stride?

Will you still listen to them,
who claim they are divine?
people who lead you astray,
one wrong step at a time?

Will we cease our endless fighting,
over something we didn't choose to be?
not just stuck with our identity,
but strive for better humanity.

In a world devoid of deities,
will we find our moral compass clear?

Or will we proceed in the ages,
which is fuelled by ignorance and fear?

When the pillars of faith crumble,
will we rise or will we stumble?

Will we get a chance to see,
a new world built on true empathy?
With our hearts united, we might find,
a brighter and promising future for mankind.

Maybe if we live like "God is Dead",
we might start doing things sensible,
or maybe God is already dead,
and we are the ones responsible!

-Yubaraj Das
3rd year

ROUGH ROADS

Beginnings are vibrant,
we dream of wrapping it up as fast
But no one tells you that roads are rough,
disrupted with tons of illusions and dust.
We go strong, we fall, compete and run
Journey is yet everlasting, it's never 'done'.
Stuck in a hollow, we need pace
But the clock waits for none
Didn't goals seem right in front of our face,

once we had begun?

Trounced with the idea, we never know what to blame,
Our wrath is irresistible, relentlessly questioning this fate.

What if roads to finding ourselves
ever lead to answers being searched for decades?
So here we rise again, do crawl but never stop
Out of the hollow now, racing with the clock
Standing resilient, we accept it all.
Undisturbed, where the 'end' is,
this journey is immensely propelling.
Our minds end up in tranquil peace.

-Avilasa Maji
3rd year

THE MAN ON THE MOON

I was reading a book when I looked up. There at the window I saw his shadow. He claimed to be the Man on the Moon. But I knew better. Why wouldn't I? My mission calls me at all times. The world was for me to save. The intricate balance of Life and Death swirled around like opposing storms before my eyes. I had no choice but to follow my destiny. He had armours and shields, I required none. Soon we reached the inevitable prison of terror. Rumors had it that only I could defeat the Cyborg. I had no room for errors or self-doubt. Fire was running through my veins.

It came out from the dark. It's first sight didn't surprise me but it's infinite powers sure did. In no time I felt tired and defeated, but my heart and brain weren't synonymous in their decision. They refused to accept defeat. This was the time when I discovered the source of the turmoil, the source that would either be my victory or fateful peace. I blasted the only sources of illumination deeming the place into absolute darkness. While it struggled to trace my existence, I extinguished it's. Pulling out the energy source that supplied power to it, I collapsed on my knees. The Man on the Moon knew better indeed.

-Anuska Roy
2nd year

जुगनू

कई महीनो बाद, मैं खुद को समेटा हूं,
आज मैं इत्मीनान से खुले आसमान में बैठा हूं
ये दिन से साल हो गए और साल से कई साल हो गए,
ये आसमां जैसे, मेरी तरह रुकी हुई है,
और ये हवाएं जैसे, समय की तरह जा रही है,
ये इमारत जैसे, मेरी जिम्मेवारियों का याद दिला रही है
ये तारें जैसे, मेरे दोस्तों सा रुख-मोड़ रही है
और ये चांद की दाग, न जाने किसकी याद दिला रही है
लो ये बारिश की बूंदे, हथेली भी चूम ली
बादलों को भी कहां पसंद ये दो पल की खुशियां,

ये बारिश की बूंदे जैसे,
शुरू के दिनों में पापा का पैसा भेजना याद दिलाता है,
लेकिन धीरे धीरे वह,
एक- एक पैसा का दिल दहल जाता है।

हां, वो बारिश के बूंदों का,
मिट्टी को चूमना ,
थोड़ा सा धैर्य रखना सिखाता है
हां घर से निकलते वक्त
ठेकुआ- पिरिकिया के साथ,
मां का चरणवदना याद आता है।

लेकिन इन दिलाशाओ के बीच,
उनका उम्मीद कहां खो पाता हूं मैं,
जब भी उनकी याद आती हैं,
थोड़ा सा रो लेता हूं मैं।

ये नुक्कड़ का कीचड़ जैसे, मन है मेरा,
क्या सब मिला है इसमें, कहां पता चल पाता है
ये कुर्सी और रस्सी तो हमारे यहां भी है
लेकिन ये घर की जिम्मेवारी, मन साफ कर जाता है।

ये कहानी छोटी सी मकसद की हैं
और ये मकसद कहां आसमां हैं मेरा,
बस बनना है वो उड़ता हुआ जुगनू सा
जिससे थोड़ा सा, रोशनी युक्त घर हो जाएं मेरा।

-Ranjan Kumar
4th year

সংস্কার

কান পাতলেই শুনি ডাকছে পরাজয়,
অথচ, রক্তে আছে বিজয়ী হবার নেশা!
একটুখানি সামনে গেলেই হাতছানি দেয় জয়,
নষ্ট বোধের ক্লান্তির অজুহাতে পশ্চাতে ফিরে আসা।

যত দোষ সব ভাগ্যের; অবাধে আওড়ায় বুলি,
ছত্রাক খেলে পঁচা মগজে চলে কু-বুদ্ধির ডাঙ্গুলি।
হেলাফেলা করে সময় গড়ায় মনোযোগী অকর্মে,
বোধ হয় না কর্ম করার চর্বি জমেছে চর্মে।

সুবোধ আর ওঠে না জেগে নির্বোধ ভালোবাসে,
আলো উঠলেই সুবোধ ঘুমায়, দুর্বোধেরা হাসে-
বের হলেই নতুন কিছু খামচে ধরবে পা,
অতি আদুরে মনুষ্য জীবনের মাথায় পরবে ঘা।’

ফিরে আসি সব গ্লানি ভুলে নিজ মৃত বোধের কাছে,
কত অজানা গল্পের সফল সমাপ্তি অযত্নে গঁথে আছে।
চমকে উঠি নিজেকে দেখে বোধ হারালাম নাকি?
অহেতুক এই অন্যায় বাঁচা নিজেকে দিয়েই ফাঁকি।

মৃতবোধ আজ হেসে ওঠে আমাকে জীবিত দেখে,
জয়ী হও নয়তো মরে যাও- নইলে মানুষ কিসে!
সুবোধ জাগে না, জাগাতে হয়- এই-ই শ্রেষ্ঠত্বের ধারক,
জাগিয়ে তোলো মৃত সুবোধ, হয়ে ওঠো সংস্কারক।

বৃদ্ধাশ্রম

বসে আছি একদিন সন্ধ্যা বেলায়
হঠাৎ যেই ছেলেটি দেখা -শোনা করে সে ছুটে এলো
হাতে তার মোবাইল ফোন ,
ফোনটা আমার হাতে দিয়ে বললো
“মাসিমা তোমার ছেলেকে দেখবে দেখবে করছিলে না এই দেখো ”
তাকিয়ে দেখি সত্যি আমার খোকা
আগেশুনেছি সে নাকি বিশাল মোটিভেশনাল স্পিকার হয়েছে ,
অনেক নামতার ,
তার সেইদিনের বিষয় ছিল “মা বাবার প্রতি সন্তানের কর্তব্য ”
সত্যিই দারুন কথা বলতে শিখেছে আমার ছোটখোকা
মনছুঁয়ে গেলো তার প্রতিটিকথা
বক্তব্য শেষ করার আগে খোকা বললো সে নাকি তার মাকে অনেকদিন আগেই
হারিয়েছে
এইশেষ কথাটা শোনায় কেমন যেন স্তব্ধ হয়ে গেলো চারিদিক
মনেহলো বুকের ওপর যেন ভারী পাথরচাপিয়ে দিলো কেউ ,
আমারপাশে বসে থাকা ছেলেটি কিছু না বলেই ,
ফোনটাবন্ধ করে আমায় বললো
“মাসিমা এবার চট করে ওষুধটা খেয়ে ঘুমিয়ে পর দেখি “
তাকিয়ে দেখি তারও চোখে জল

-Kushal Mukherjee
4th Year

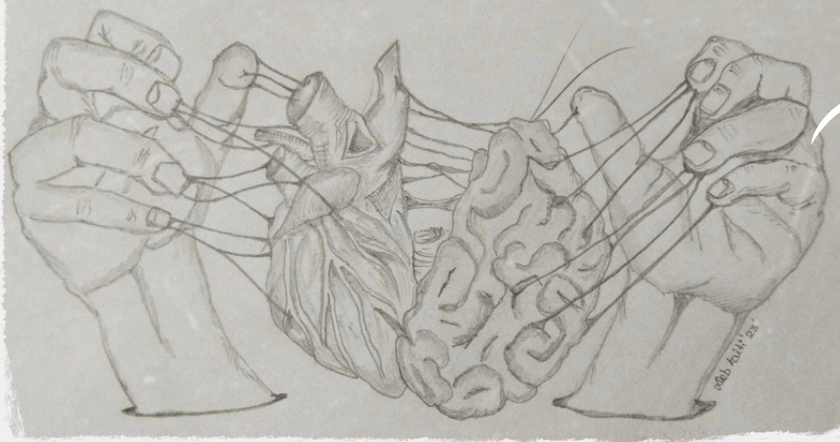
A close-up photograph of a hand holding a paintbrush, poised over a canvas. The canvas features abstract, textured brushstrokes in shades of green, blue, and white. The hand is wearing a ring on the ring finger. The entire image is framed by a double-line border in a light beige color.

COLOUR
conversation

Shreya Denre
2nd Year

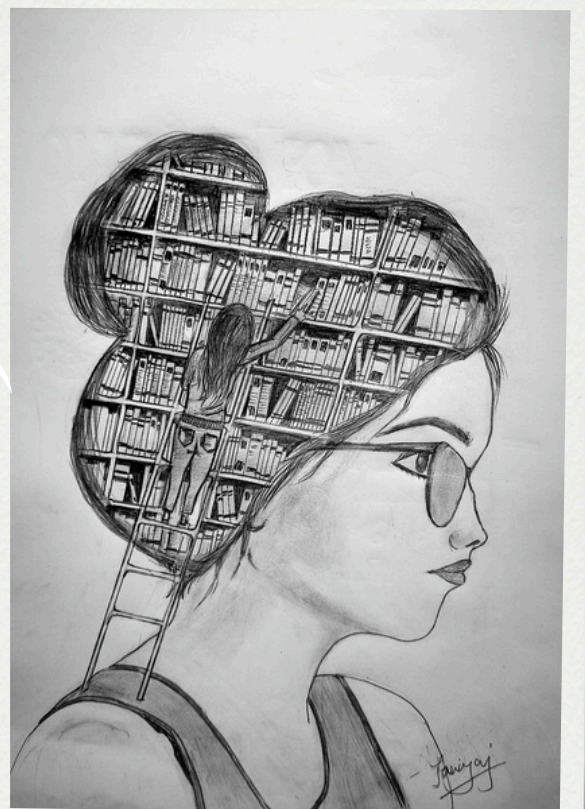


Shreya Denre (CSE B)
2nd yr.



Deb Kaiti
4th Year

Tani Raj
3rd Year

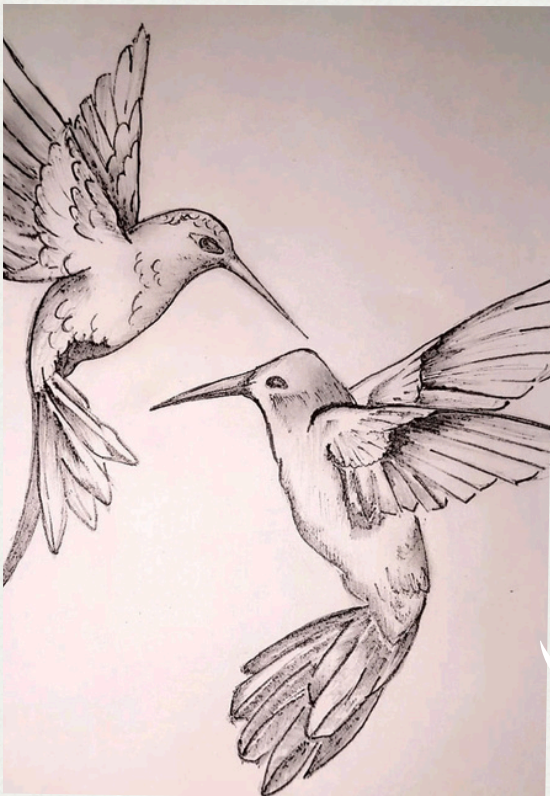




Mahek Agarwal
4th Year



Swapnil Chaki
2nd Year



Anushka Nath
3rd Year

Kritika Singh
3rd Year



Ramyani Das
3rd Year



Saheli Kundu
2nd Year



victory vault





WINNER

Event: **eXabyte 2024 - Annual Technological Festival**

Venue: St. Xavier's College (Autonomous), Kolkata

Date: 12th & 13th of February, 2024.

Team Members:

- **Yash Goyal(4th year)**
- **Mahek Agarwal(4th year)**

- Represented Heritage Institute of Technology in the flagship event **"CodePlay"**
- Blind coding competition wherein both the team-member had to write the code with the monitors switched-off.
- The competition checked both the typing skills and correctness of the code by checking test-cases.



WINNER

Team name: **BinaryBrainz**

Event: **HackNITR**

Venue: **NIT Rourkela**

Position: **1st**

Team Members:

1.Sagnik Pramanik (TEAM LEAD)

CSE-C 2026

In charge of the ML and Processing

2.Muskan

CSBS 2026

In charge of the UIUX design and partly the frontend of the website

3.Suparnojit Sarka

CSBS 2026

In charge of the backend for the website

4.Sauhardya Chakraborty

CSE-A 2026

In charge of the frontend

Project Name: DeHazeXpert

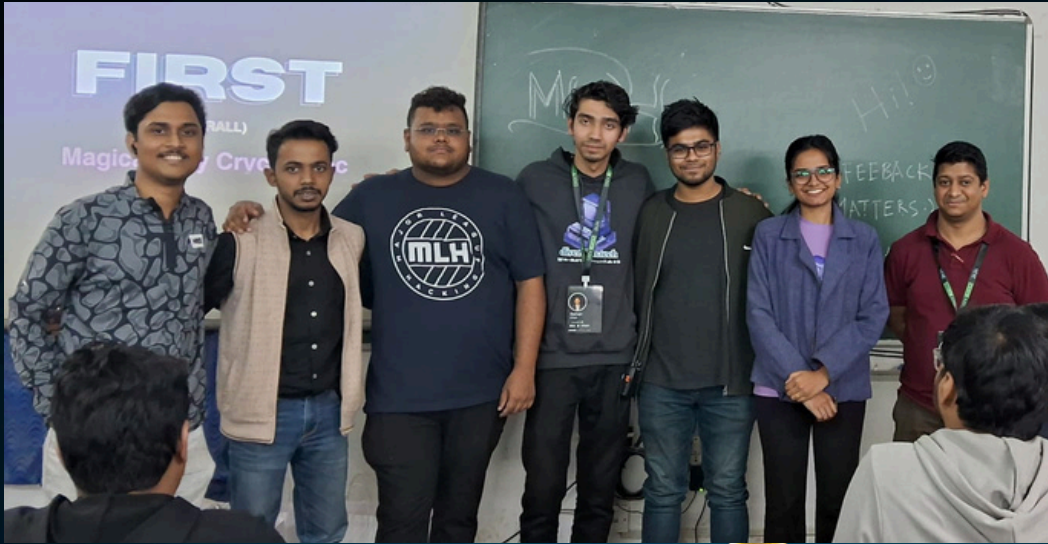
Real-Time Video enhancers (make the video clear by removing haze/ smoke) with fire detection, tagging, and tracking human detection and tagging.

Unique Thing about the project:

Image enhancers already exist in the market but we are the first to do it with video and IN REAL TIME. (No one has done that before.) The latency is about 0.2 seconds which is very low.

- > More than 3000 people applied
- > After profile screening round, 80 teams were selected and invited to NIT Rourkela for the in-person hackathon
- > After judging rounds by mentors, the top 12 teams were chosen. These top 12 teams had to pitch their project ideas In front of another set of judges, consisting of eminent people.





WINNER

Name: **Anurag Dutta**

Dept. : CSE

Sec : A

Year : 4th

- Secured the 1st Position at **DIVERSION** 2K24, a 36-hour offline hackathon organized by IEM Kolkata on February 3rd and 4th, 2024.
- The hackathon was supported by MLH (Major League Hacking) and had a maximum team size of 4 members.
- Participated as a team of four, along with three friends from different institutes.



- The project, **CryoSync**, is an IoT-based solution designed for temperature-specific environments such as nuclear reactors and cold storage facilities.
- CryoSync enables the storage and transmission of temperature or threshold values from these challenging environments to a centralized administrator, where human intervention is difficult due to extreme temperatures.



Legacy

SPOTLIGHT



“Hello everyone, I am **Jibitesh Chakraborty**, from the 2024 batch of B.Tech CSE Department, and I have been asked by my juniors to write something insightful from my 4 year journey at Heritage Institute of Technology and coding in general.

Coming to my college life, my 4 years have been a roller coaster ride, with ample of highs and lows, yet never failing to keep me engaged and thrilled all along. Apart from the immense knowledge I gathered from each and everyone at the CSE Department, what I really feel is a gift was my interaction with different departments. I have learnt a lot from my batchmates, seniors, and most importantly my juniors. Sounds surprising right, but yes, I genuinely have learnt a lot from my juniors, both technical and non-technical.

Coming to how one can be utterly comfortable while appearing for placement drives, both on-campus and off-campus. This is something that we all know but fail to implement in real life. So, let me just reiterate what we need to do. First and foremost, keep your fundamentals clear. Do not be someone who knows everything yet nothing. Subjects like Operating Systems, Computer Networks, Object Oriented Programming and Database Management Systems, are going to stay with you for a lifetime. Keep these subjects brushed up in your mind, all the time.

Data Structures and Algorithms is of prime importance. Nobody is asking you to be a Professional Competitive Programmer, but you genuinely need to practice problems from Leetcode on a regular basis. Leetcode has a particular set of questions called the “75 Blind Post Questions”, solve them diligently. That’s your bible to any coding interview. Choose one language for DSA, preferably Java8 or C++, and learn everything, A-to-Z in DSA in that specific language. Trust me, coding rounds are not that difficult.

Now, DSA is also not everything, your resume too should have some element in it. Choose at least one, preferably two, technologies or domains to make projects on. My choices were ML/DL and Web Development. So, for my projects, I focused only on python [Tensorflow] for ML/DL projects and Javascript [React+Node] for my Web Development projects. You can choose from a wide variety of domains, with Blockchain and Cybersecurity too in the list. Keep your resume as clean as possible. Use neutral colors in your resume and use as many keywords as possible while mentioning your skills or describing your projects. Resume needs to be as much ATS friendly as possible.

And lastly, remember, rejections are the stepping-stones to success and the company that rejects you is not the last company in this world. Your career is a marathon and not a sprint. So take your time, that does not mean you just become lazy. Be consistent with whatever you do and never ever compare yourself with your peers. Just stay motivated and keep your confidence high. Be confident, but never overconfident. And that’s it, you are placement ready. I am looking forward to seeing incredible engineers, graduating from Heritage year after year. Keep your spirits high and your mind eager to learn. I wish the very best for each one of you. Cheers to you all!

-Jibitesh Chakraborty
Scholar, SAP
Class of 2024





Hello everyone, I'm **Srinjoy Ray**, a 2024 Computer Science and Engineering graduate from Heritage Institute of Technology. I'm currently working at **Google**, and before this, I had the opportunity to intern at **Flipkart**. Through these experiences, I've gathered a lot of insights and knowledge about preparing for and succeeding in big tech interviews. In this article, I'm going to share some of the most relevant tips and strategies that can help you ace your interviews at major tech companies.

How to Crack Tech Interviews

Understanding the Interview Process

So, are you ready to dive into the world of tech interviews with giants like Google, Amazon, and Microsoft? Buckle up, because here's a peek into the rollercoaster ride that awaits:



Finding Job Openings

Think of your job search like a treasure hunt. Check company websites often to find job openings that match your skills. When you find one, apply quickly! If you can get a referral from someone who works there, it's like having brownie points.

Initial Shortlisting

Once you apply, your resume will be reviewed by recruiters during the shortlisting process. This can be a bit of a gamble—sometimes even great resumes get overlooked. If that happens, don't get discouraged—keep applying to other opportunities. If you do get shortlisted, give yourself a pat on the back! Now it's time to bring your A-game.

Online Assessment

The first step in the actual interview process is often an online coding test. This is usually a timed test where you'll need to solve problems related to algorithms and data structures.

Technical Interviews

If you pass the test, you'll have a series of technical interviews. These focus on your coding skills, problem-solving, and computer science basics. It's your chance to show off your tech know-how!

Behavioural Interviews

In addition to technical interviews, you might also face a behavioural interview. This is your chance to shine a light on your life experiences. You'll discuss how you've handled various situations in the past, including team projects and problem-solving scenarios. Use these stories to show that you're not only a coding whiz but also someone who fits well with their team's culture.

Preparing for the Interview



Master Key Data Structures

- **Arrays & Strings:** Get comfortable with basic operations like adding, deleting, and searching—they're the bread and butter of many problems.
- **Linked Lists, Stacks & Queues:** Understand their concepts, real-world applications, and how they're commonly used in various problems.
- **Trees & Graphs:** Grasp binary search trees, traversal techniques, and graph algorithms like DFS, BFS, and Dijkstra's.
- **Hashing & Heaps:** Learn hash tables, collision handling, and binary heaps for efficient data management.

Strengthen Algorithmic Skills

- **Sorting & Searching:** Know the sorting algorithms and binary search like the back of your hand.
- **Dynamic Programming:** Master breaking down problems into subproblems, a crucial skill in complex problem-solving.
- **Greedy, Recursion & Backtracking:** Practise these techniques regularly. They're crucial for solving problems that require optimization.

Consistent Practice

- **Regular Coding Practice:** Dedicate time to solving problems on platforms like LeetCode, beginning with simpler problems and gradually advancing to more complex ones. Participate in as many contests as possible since that can mock a real interview environment.

Solidify Core Concepts

- **Computer Science Fundamentals:** Make sure you have a strong grasp of computer science fundamentals like operating systems, databases, and networking. They're key to nailing technical interviews.

During the Interview

Communicate Clearly

Interviewers want to see how you think, not just your final answer. Explain your thought process as you work through problems. It's like showing your rough work in maths class.

Ask Clarifying Questions

If a problem statement isn't clear, don't hesitate to ask questions. It helps you avoid mistakes and shows that you're thorough.

Manage Your Time

Keep an eye on the clock and make steady progress. If you get stuck, move on to another part of the problem and come back later. Time management is key!

Stay Calm

Interviews can be nerve-wracking, but staying calm is crucial. Take deep breaths and focus on one step at a time. Staying cool will help you tackle those tough questions with ease.

Conclusion

Interviews can be an exciting journey, and with solid preparation and a positive attitude, you'll be on track to landing your dream job. Best of luck, and don't forget to have fun during your preparation journey!

For more tips and insights, feel free to connect with me on LinkedIn: www.linkedin.com/in/srinjoy-ray-874b031b6

-Srinjoy Ray
Software Engineer, Google
Class of 2024



Hello to all you brilliant minds in the Computer Science department! As an alumnus who once sat in the same lecture halls and battled the same tricky algorithms, I'm thrilled to have this opportunity to reconnect and share a few insights from the other side of graduation. Think of this as a friendly guide, or if you prefer, a survival manual for the vast hurricanes, tornadoes and sometimes chaotic world of software engineering. So, buckle up—let's dive into the world of software development, with a few laughs along the way.

Navigating the Digital Frontier : An Alumni's Guide to Jumpstart Your Software Engineering Career

Master the Basics: The Foundation of Your Software Empire

Let's start with the fundamentals. Imagine you're building a house. You wouldn't start with the roof or the wallpaper; you'd begin with a strong foundation. Similarly, in software engineering, mastering the basics is crucial. Think of algorithms and data structures as your house's foundation. Without them, your software is just a fancy-looking pile of code ready to crumble at the first sign of stress.

Data Structures and Algorithms: Algorithms are the recipes for solving problems. If you've ever followed a cooking recipe, you know how important it is to get it right. The same goes for algorithms. A good algorithm will make your code more efficient and help you avoid that dreaded "Out of Memory" error. On the other hand, data structures are like the containers that hold your data. If you're organising a party, you wouldn't put everything in one giant box. Instead, you'd use different containers for different items: one for drinks, one for snacks, and so on. Data structures are your containers—arrays, linked lists, trees, and graphs—each designed for specific types of tasks.

Explore your Kingdom: The Realm of Software Engineering domains

Imagine yourself to be the king of your kingdom, If you do not study the territories and provinces of your kingdom well - you wouldn't be able to expertise yourself in understanding which area demands more attention and where you should focus on increasing your kingdom's resources.

On a similar note, start exploring the different realms and domains of software engineering from the early phases of your software development journey. You may start off with front-end development as your initial interest, but may end up being a DevOps specialist which provides your more satisfaction in solving problems across the terrains of software engineering. Why is this so important? In my past experience and encounters with junior folks from different colleges, I've found people remain confused as to what domain they should proceed with. This usually comes up, where either the campus placements knock the door or sometimes a youtube playlist has caught your attention. Thus, starting early and understanding the bits and bytes of every domain is crucial to steer you forward in this journey.



Hands-On Experience: Get Your Hands Dirty

Theory is great, but practice makes perfect—or at least less of a mess. Dive into projects that interest you. Build an app, create a game, or even develop a new tool that solves a problem you've encountered.



Personal Projects: Learn and learn using Google and YouTube. These two entities will be your best friends in your entire software engineering journey. But, only learning will not make the pizza be served on your table. You need to implement your learning in an innovative and effective way. Often, I have seen people following a playlist for learning purposes and just put that up as a personal project. My friend, refrain from doing this at all cost ! Untie the knots of your brain cells and nerves and build something, that's solely your own contribution from the start and till the end.



Open Source Contributions: Want to feel like a superhero? Contribute to open-source projects. You get to work on real-world problems, collaborate with other developers, and maybe even fix that one bug that's been driving everyone crazy. Plus, it's a great way to make your mark in the developer community. I highly recommend having an active open source contribution profile. You can start off initially with GirlScript Summer of Code and then step up towards larger open source programs like The Linux Foundation Mentorship and Google Summer Of Code. This makes you stand out from the rest of the crowd.

Lifelong Learning: Staying Ahead in the Game

Stay Curious: Follow tech blogs, join online forums, and read up on industry news. Websites like TechCrunch and Medium are excellent for catching up on the latest trends. It's like being part of a very tech-savvy, slightly geeky club. Reading a pdf or book to gain insights on the theoretical and practical aspects of software engineering is much more difficult as compared to plugging in your earphones and watching a 5 minute video. Once you master this habit, there's no looking behind.

Attend Conferences and Meetups: Conferences are like tech-themed amusement parks. You get to meet industry experts, learn about cutting-edge technologies, and perhaps even grab a swag bag. Local meetups are a great way to network and find out what's buzzing in your area.

Communication and Teamwork: The Soft Skills You Need

In software engineering, communication is as important as coding skills. You'll need to explain your ideas, collaborate with others, and sometimes even decipher cryptic bug reports from colleagues.

Clear Communication: Whether you're drafting documentation, writing code comments, or discussing project requirements, clarity is key. Avoid jargon overload and make sure everyone on your team understands your vision. Remember, "It works on my machine" is not an acceptable status update.

Collaboration Tools: Get comfortable with tools like Git for version control, JIRA for project management, and Slack for team chat. These tools help keep your projects on track and your team connected. Think of them as the digital equivalent of coffee breaks—essential for maintaining sanity.

Conclusion

As I look back on my own journey, I'm filled with gratitude for the experiences and lessons learned. I hope these insights and tips help you as you embark on your own path in software engineering. Embrace the basics, get hands-on, stay curious, and remember that communication and resilience are as crucial as your coding skills.

Keep experimenting, keep learning, and most importantly, enjoy the ride. The world of software engineering is vast and full of opportunities. So go out there, make your mark, and remember: every line of code is a step towards creating something amazing.

Wishing you all the best on your journey—may your code be bug-free and your coffee always hot !

-Satabrata Paul
Software Engineer II, Atlan
Class of 2023

STUDENT INNOVATION

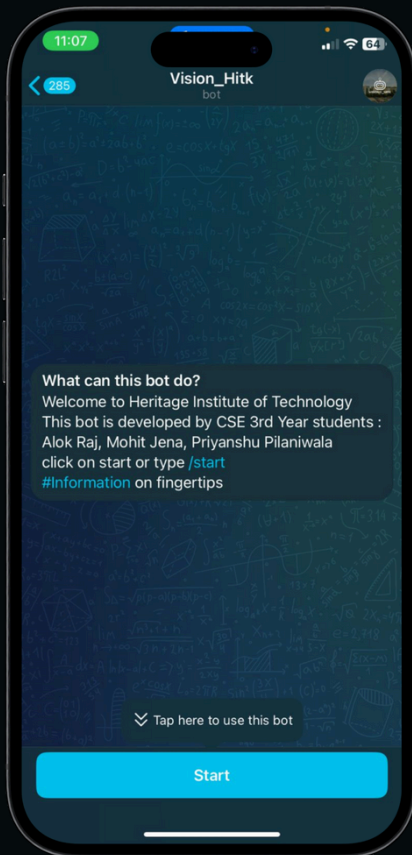




VISION.0_HITK

Apke Har Sawal ka Jabab

@alok@mohit@priy



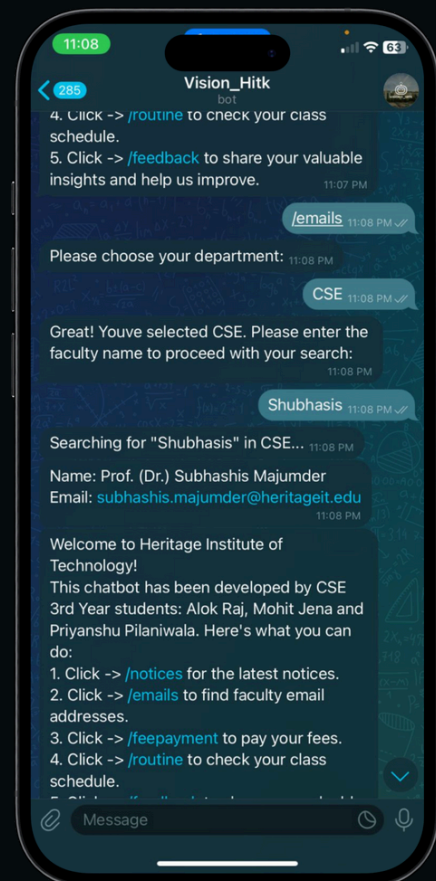
Introducing **VISION**: Your Personal College Companion

Navigating college life just got a whole lot easier with VISION — the smart chatbot designed to simplify your day-to-day activities on campus. Imagine having a virtual assistant that not only keeps you updated with the latest happenings but also connects you directly with faculty members and guides you to your next class. That's exactly what VISION offers!

What Does VISION Do?

1. **Instant Faculty Contacts** : Need to reach out to a professor? VISION provides you with the email IDs of faculty members in seconds, enabling quick communication and reducing the time spent searching through directories or websites .

2. **Stay Updated with Notices** : Forget missing out on important announcements or notices pinned to bulletin boards. VISION delivers the latest college updates directly to you, so you are always in the loop.

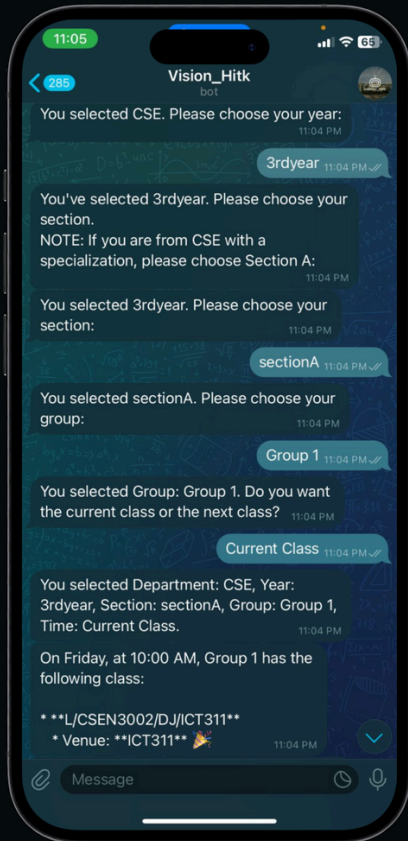




VISION.0_HITK

Apke Har Sawal ka Jabab

@alok@mohit@priy



3. Find Your Class Effortlessly : No more rushing through corridors or asking multiple people for directions! Whether it's your next class or your current lecture location, VISION provides clear, precise directions, saving you time and hassle.

How to Access VISION?

VISION is available to all students and faculty members via the telegram platform. Simply search 'Vision_hitk' and you're ready to explore the features that make your college experience smoother and more efficient.

Say goodbye to the stress of finding information or navigating campus life. Embrace VISION— your all-in-one solution for a seamless college experience. Make your life easier, more organized, and stay ahead with VISION.



Mohit Jena
3rd year



Priyanshu Pilaniwala
3rd year



Alok Raj
3rd year

INSIDE SCOOP



WINNER
Inter-Department
Best Department
CSE





MEET OUR STUDENT & EDITORIAL BOARD



Sneha Raj
3rd Year



Ankita Sharma
3rd Year



Kritika Singh
3rd Year



Nausheen
3rd Year

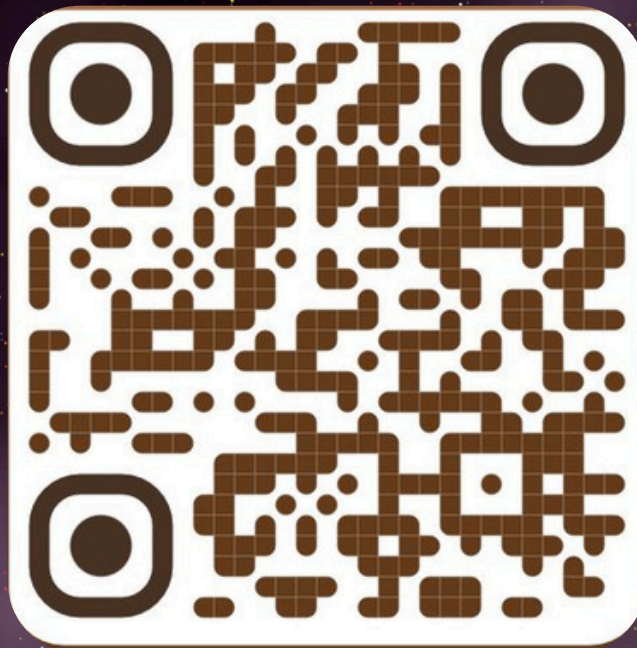


Samraj Banik
4th Year



Sonajhoori Maitra
4th Year

*Thank
You!*



**SCAN QR
TO GIVE FEEDBACK**