NATIONAL EDUCATION SOCIETY

J.N.N College of Engineering,

Shimoga-577204

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Cosmobyte 22-23

MESSAGE FROM HOD

Dr. Poornima K.M

Dear Students Greetings,
I feel proud and honored to lead such an active and energetic department. The major strength of the department is a team of well qualified experienced and dedicated faculty who are continuously working and supporting the students for their excellence in academics, sports and cultural activities.

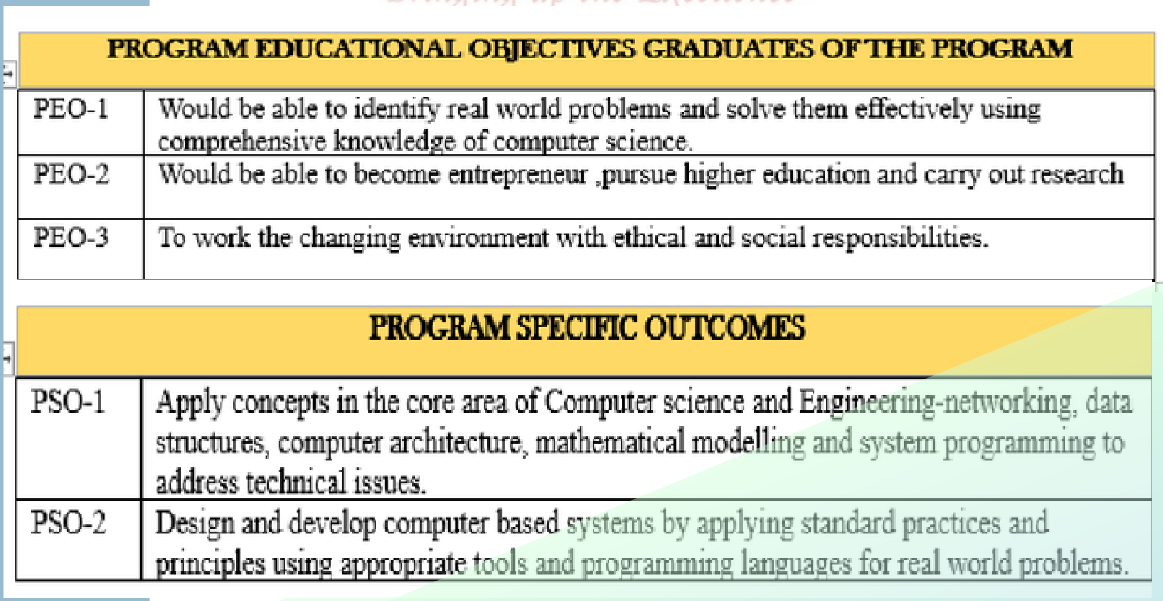
I am very happy that, our CS&E department is releasing annual magazine COSMOBYTE. This magazine gives an insight into the range and scope of the imagination and creativity of our students My heartfelt congratulations to the entire team for their efforts in bringing out the department magazine.
No one reaches a peak of the hill just by looking at one who has determination and stamina and to go through the total toil will reach the pinnacle. My dear students I wish all the very best for your future endeavors.

VISION

To be one of the pre-eminent departments to provide technical and knowledge based education, utilizing the potential of Computer science and Engineering to meet the ever changing needs of society and industry.

MISSION

- Mold the students to meet the emerging challenges of industry and society.
- Emphasizing on research.
- Effective industry interaction for the development of state of the art technological infrastructure and faculty component.



PROGRAM EDUCATIONAL OBJECTIVES GRADUATES OF THE PROGRAM	
PEO-1	Would be able to identify real world problems and solve them effectively using comprehensive knowledge of computer science.
PEO-2	Would be able to become entrepreneur ,pursue higher education and carry out research
PEO-3	To work the changing environment with ethical and social responsibilities.

PROGRAM SPECIFIC OUTCOMES	
PSO-1	Apply concepts in the core area of Computer science and Engineering-networking, data structures, computer architecture, mathematical modelling and system programming to address technical issues.
PSO-2	Design and develop computer based systems by applying standard practices and principles using appropriate tools and programming languages for real world problems.

EDITORIAL BOARD

CHEIF EDITORS
Ms.Sinchana S Noolee,8th Sem
Ms. Shubha M.L,8th Sem

SUB EDITORS
Mr.Nandish D,6th Sem
Ms. Pooja B,6th Sem

VERSE AND VOICES

I NEVER KNOW WHY,
WHY AM I BEING ME..
LET THE TIME FREEZE
FOR ME TO CHERISH YOU

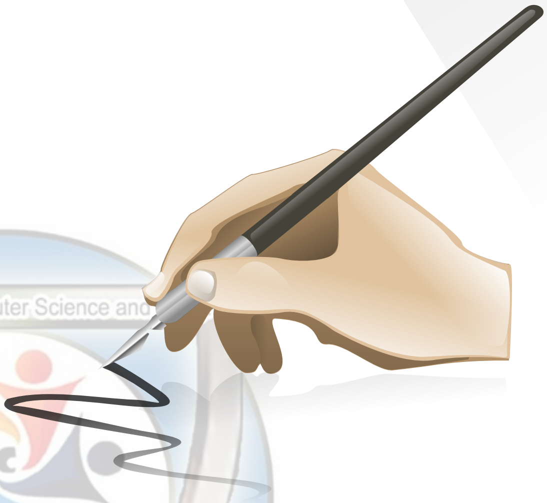
SMILE IN YOUR EYES
STEALS MY SILENCE
UNSEEN VOICE OF YOURS
SEIZE MY WORDS

THOSE HEARTFELT LINES
WHICH REMAIN UNSAID
THOSE CRAZY THOUGHTS
TO BE AROUND YOU
THE WAY I FEEL YOU
I SWEAR NO ONE CAN

GIVE ME A MOMENT
I'M COMPLETELY IN YOU
JUST LOOK AT ME
FOR ONCE AND I'M DONE

NOTHING ,JUST A WISH
I'VE BEEN WAITING FOR
JUST SAY YES
AND I'LL NEVER REGRET

-SHUBHA M L, 8B



PEOPLE ASKS WHY SHE LOVES
SUNSET
IF THEY MEAN THAT EVERYTHING
HAS THEIR END.

BUT THE COLOR OF THE SUN.
IS WHAT MAKES HER HAPPY.

IT REMINDS HER
THAT WHATEVER HAPPENED
THE ENTIRE DAY.
YOU CAN STILL GET
A BEAUTIFUL THING
AT THE END

-POOJA R, 6B

ಮುಸ್ಸಂಜೆ ಮುಸುಕಿನಲಿ
ಏಕಾಂಗಿ ಮನವ ಹೊತ್ತು
ಹೊರಟೆನು ಎಲ್ಲಿಗೋ..
ಕಾಡುತಿಹ ನಿನ್ನ ನೆನಪ ನೆಪದಲ್ಲಿ
ಸೋತಿಹುದು ಮನ ಮೌನದಲ್ಲಿ

ಆಡುವ ಮಾತಿರಲು ಹಲವು
ಕಾಣುವ ಕಂಗಳು ಸೋತಿಹವು
ದೂರದ ದಾರಿ ಸವೆದಿರಲು
ನಿನ್ನಯ ನೆನಪೆ ಜೊತೆಗಿಹುದು

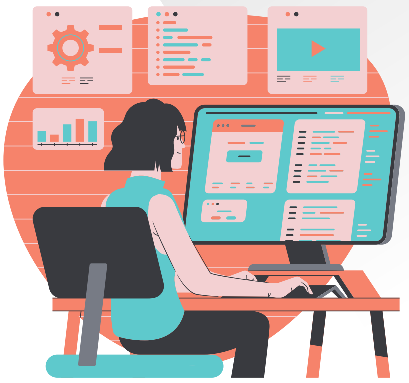
ಬೀಸುವ ಗಾಳಿ ಅಪ್ಪಿರಲು
ಬಯಸಿದೆ ನಿನ್ನೇ ಸೇರಲು
ಕತ್ತಲೆಯ ದೀಪ ಹೊತ್ತಿರಲು
ಭರವಸೆಯ ಬೆಳಕು ಮೂಡಿಹುದು

-SHUBHA M L, 8B

ಮೊದಲಸಲ ತಿಳಿಯದ ಕನಸು ನೀನು
ಪದೇ ಪದೇ ಕಾಡುವ ಕಲ್ಪನೆಯು ನೀನು
ನಿಘOಟಿನಲ್ಲೂ ಇಲ್ಲದ ಪದಗಳ ಅರ್ಥ ನೀನು
ನನಗೆ ಸಿಗಬೇಕೆನ್ನುವ ಸ್ವಾರ್ಥ ನೀನು
ಪುನಃ ಕಲಿಯಬೇಕಾದ ಕಾಗುಣಿತವು ನೀನು
ಸೌಂದರ್ಯ ದೇವತೆಯ ಅಂಶ ನೀನು
ನಿದ್ದೆಯಲ್ಲೂ ಕಾಡುವ ಕ್ರೂರಿ ನೀನು
ಮುದ್ದು ಮುದ್ದಾದ ಕೃತಿಯು ನೀನು
ಕಾಳಜಿ ಇಂದ ಕೆತ್ತಿದ ಆಕೃತಿಯು ನೀನು
ಮನಸ್ಸಿನ ಭಾವ ನೀನು
ಹೃದಯದ ಬಡಿತವು ನೀನು
ಈ ಕವಿತೆಯ ಬರೆದಿರುವುದು ಕೂಡ
ನಾನಲ್ಲ,ನನ್ನೊಳಗಿನ ನೀನು

-MANJUSH V, 8A

ಮುಗಿಯಿತು ಪದವಿ ಪೂರ್ವ,
ಯೋಚ್ನೆ ಮಾಡಿದೆ ನಾನು-ಇಂಜಿನಿಯರಿಂಗ್ ಸೇರ್ಕೊಳದ ಬೇಡ್ವಾ,,
ಅಂತು ಇಂತು ಸೇರಿದೆ,
ಏನೇನು ಯೋಚ್ನೆ ಮಾಡದೇ,,
ಸ್ಟೆಪ್ ಬ್ಲಾಕ್ ಅಲ್ಲಿ ಫಸ್ಟ್ ಇಯರ್,
ಆಮೇಲ್ ಕೊರೋನಾ ಹೇಳ್ತು ಹ್ಯಾಪಿ ನ್ಯೂ ಇಯರ್,,
ಫ್ರೆಶರ್ಸ್ ಇಲ್ಲದೆ ಬ್ರಾಂಚ್ ಎಂಟ್ರಿ ಆಗಿದ್ ನಾವು,
ಮೊದ್ಲಿಗೆ ಅನ್ನುಸ್ತು ಇಂಜಿನಿಯರಿಂಗ್ ಸಾವು,,
ಹೋಗ್ತಾ ಹೋಗ್ತಾ ಜೂನಿಯರ್ ಸೀನಿಯರ್ ಅನುಬಂಧ,
ಮನಸ್ಸಿಗೆ ಸೇರಿತು ಇಂಜಿನಿಯರಿಂಗಿನ ಗಾಳಿ ಗಂಧ,,
ಸೇರಿದ್ದು ಗೊತ್ತಾಗಿಲ್ಲ,
ಓದಿದ್ದು ಗೊತ್ತಾಗಿಲ್ಲ,
ಇಷ್ಟ್ ಬೇಗ ಈ ಪಯಣ ಮುಗಿದೇ ಹೋಯಿತ್ತಲ್ಲ,,



-VIDYASHREE B S,8 B

ಮೊದಲ ಬಾರಿ ಮೌನಿಯಾಗಿರುವೆ
ನಿನ್ನದೇ ಆಲೋಚನೆಗಳ ಸವಾರಿ
ಗುರಿಯು ನೀನೆ ಆಗಿರುವೆ
ಬಂದಿರುವೆ ನಿನ್ನ ಅಪ್ಪಣೆ ಕೋರಿ

ಶುದ್ಧ ಕನಸಿನಲ್ಲಿ ಬಿದ್ದು ಮುಳುಗಿರುವೆ
ನಿನ್ನ ಕಲ್ಪನೆಗಳ ಮನಸ್ಸಿಗೆ ತೂರಿ
ಕಷ್ಟವಾದರು ಮತ್ತೆ ಕೇಳುತಿರುವೆ
ಆಗುವೆಯ ನನ್ನ ರಾಜಕುಮಾರಿ

ಇದ್ದ ಹೃದಯವ ಹರಾಜು ಹಾಕಿರುವೆ ನೀನಾಗುವೆಯ ವ್ಯಾಪಾರಿ
ಖಚಿತವಾದರೆ ಉಚಿತ ನೀಡುವೆನು
ಎದೆ ಬಡಿತಗಳ ಲಹರಿ

ನಿನ್ನಲ್ಲಿಯೆ ಬಂಧಿಯಾಗಿರುವೆ
ನೀನೆ ಬಿಡಿಸಬೇಕು ದಯೆತೋರಿ ಮಂಡಿಯೂರಿ ಕೇಳುತಿರುವೆ
ಆಗುವೆಯ ನನ್ನ ರಾಜಕುಮಾರಿ

-MANJUSH V, 8A

IT IS ONE OF THE MOST AWAITED GEMS
SPLASHES AT THE END OF EVEN SEMS

HAS ITS GLORY WIDE SPREAD
IT'S LEGACY CAN'T GO UNSAID

GLUES TOGETHER EVERYONE
WITH EACH'S DELIGHT OVERRUN

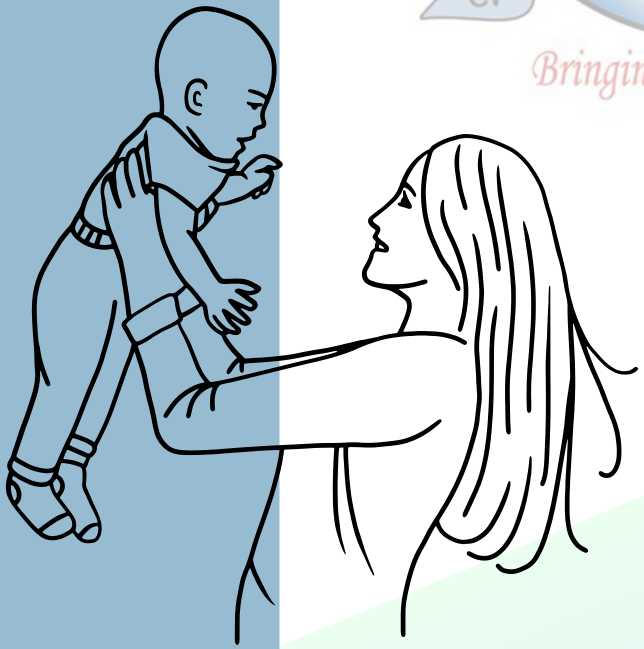
WHOLE LONG SEM WITH VARIOUS EVENTS
ALL EXECUTED WITH 100% CONFIDENCE

FILLED WITH JOY, FUN AND MEMORIES
AND HAS COVERED ALMOST ALL TRAJECTORIES

YOU ALL GUESSED IT RIGHT
WITH ALL MIGHT

IT IS THE UNBEATABLE COSMOS
WHICH ALWAYS CARRIES TREMENDOUS MOTTOS.

-PARVATHI SALERA, 8B



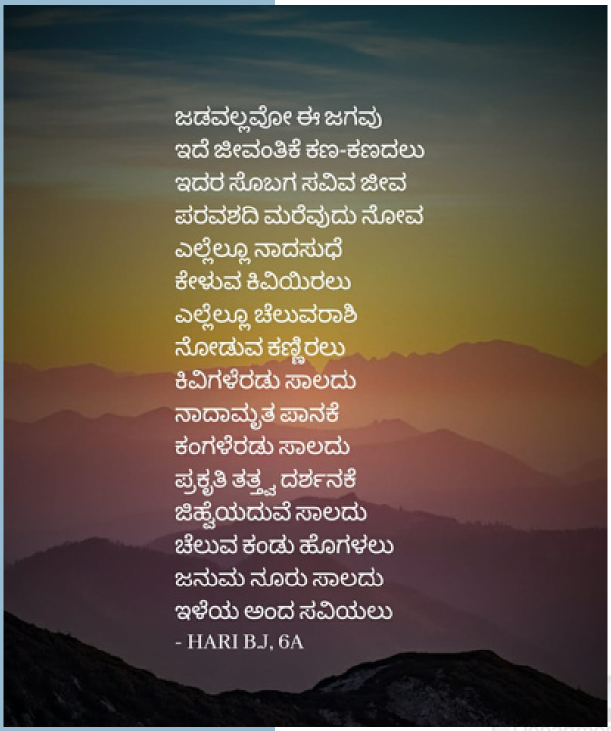
ಅಮ್ಮ

ಅಮ್ಮಾ ನನ್ನ ಅಮ್ಮಾ
ಶ್ರಮಿಸುವಳು ಹಗಲು ರಾತ್ರಿಯೆನ್ನದೆ
ಉಣ್ಣಿಸುವಳು ಪ್ರೀತಿಯ ಮಾತುಗಳಿಂದ
ನಿನ್ನ ಪ್ರೀತಿ ಆ ನೇಸರನ ಅಂಚಿನಂತೆ
ಎಂದಿಗೂ ಹೊಳೆಯುತಿರುವುದು
ಆ ರವಿಯು ತನ್ನ ಕ್ಷೇಮವ ಮರೆತಿಹನು
ಅಮ್ಮನ ಹಾಗೆ
ಅಮ್ಮಾ ನನ್ನ ಅಮ್ಮಾ.

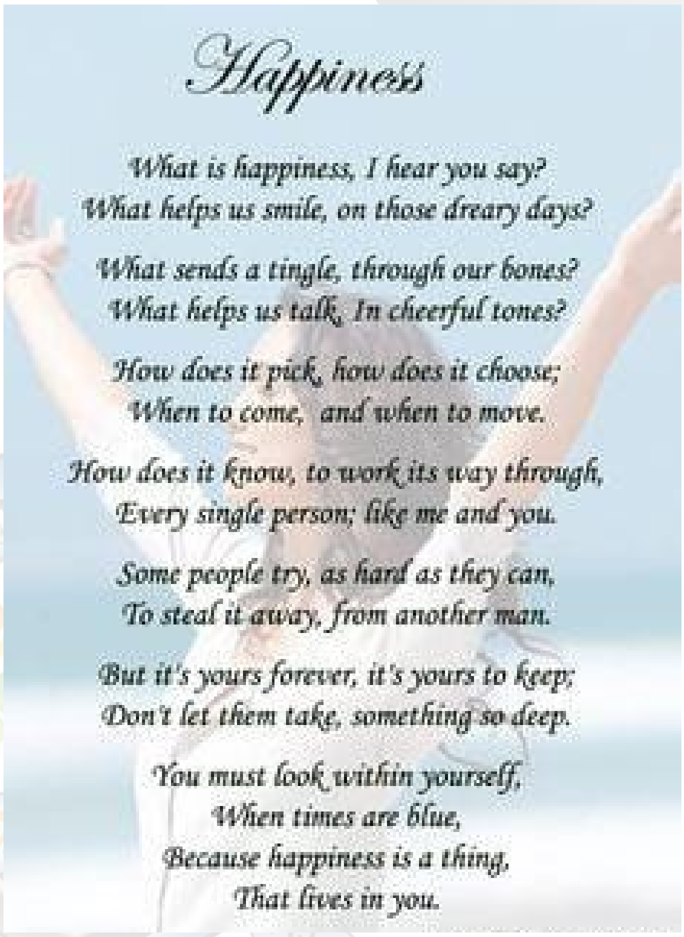
ಆ ರವಿ

ಬಾನಿನಲ್ಲಿ ಮಿನುಗುವ ನೀನು ದೂರ ಮಾತ್ರ
ನಿನ್ನ ನೆನಪಿನ ಅಂಗಳದಲ್ಲಿ ನಾನು ಮಾತ್ರ
ಮೂಂಜಾನೆ ಇರುವ ನೀನು ಸಂಜೆ ಸಿಗಲೊಲ್ಲೆ
ಮುಸ್ಸಂಜೆ ಇಲ್ಲಿ ನಾನು ನಿನ್ನ ನೆನಪಲ್ಲೇ.

-YASHASWINI I K,6B



ಜಡವಲ್ಲವೋ ಈ ಜಗವು
ಇದೆ ಜೀವಂತಿಕೆ ಕಣ-ಕಣದಲು
ಇದರ ಸೊಬಗ ಸವಿವ ಜೀವ
ಪರವಶದಿ ಮರೆವುದು ನೋವ
ಎಲ್ಲೆಲ್ಲೂ ನಾದಸುಧೆ
ಕೇಳುವ ಕಿವಿಯಿರಲು
ಎಲ್ಲೆಲ್ಲೂ ಚೆಲುವರಾಶಿ
ನೋಡುವ ಕಣ್ಣಿರಲು
ಕಿವಿಗಳೆರಡು ಸಾಲದು
ನಾದಾಮೃತ ಪಾನಕೆ
ಕಂಗಳೆರಡು ಸಾಲದು
ಪ್ರಕೃತಿ ತತ್ತ್ವ ದರ್ಶನಕೆ
ಜಿಹ್ವೆಯದುವೆ ಸಾಲದು
ಚೆಲುವ ಕಂಡು ಹೊಗಳಲು
ಜನುಮ ನೂರು ಸಾಲದು
ಇಳೆಯ ಅಂದ ಸವಿಯಲು
- HARI BJ, 6A



Happiness

What is happiness, I hear you say?
What helps us smile, on those dreary days?

What sends a tingle, through our bones?
What helps us talk, In cheerful tones?

How does it pick, how does it choose;
When to come, and when to move.

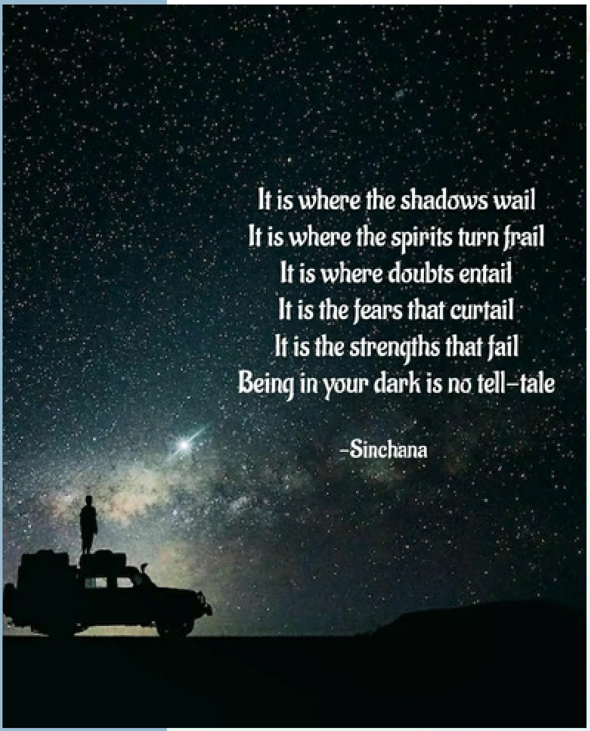
How does it know, to work its way through,
Every single person; like me and you.

Some people try, as hard as they can,
To steal it away, from another man.

But it's yours forever, it's yours to keep;
Don't let them take, something so deep.

You must look within yourself,
When times are blue,
Because happiness is a thing,
That lives in you.

-INCHARA, 6A(FROM COLLECTION)



It is where the shadows wail
It is where the spirits turn frail
It is where doubts entail
It is the fears that curtail
It is the strengths that fail
Being in your dark is no tell-tale

-Sinchana

"MY AURORA"

IN A REALM WHERE DREAMS TAKE FLIGHT,
SPLASHED IN COLORS WITH BRIGHT ETHEREAL LIGHT,
THERE EXISTS A CELESTIAL PRESENCE,
A MASTERPIECE WOVEN, BE MY AURORA.

A GRACEFUL SIGHT, AND A GENTLE MUSE,
WITH SHIMMERING OF ALL COSMIC HUES,
MY AURORA'S PRESENCE, A CELESTIAL BODY,
HEARTS AWAKENING WORLD TO A HEAVENLY BLISS.

THE CANVAS, THE ART'S FULLY IMMENSE,
A DAZZLING FEATURE SHINNING, I CONDENSE
A SYMPHONY OF SHADES, RADIANT AND RARE,
PAINTING THE HEAVENS WITH TENDER AND CARE

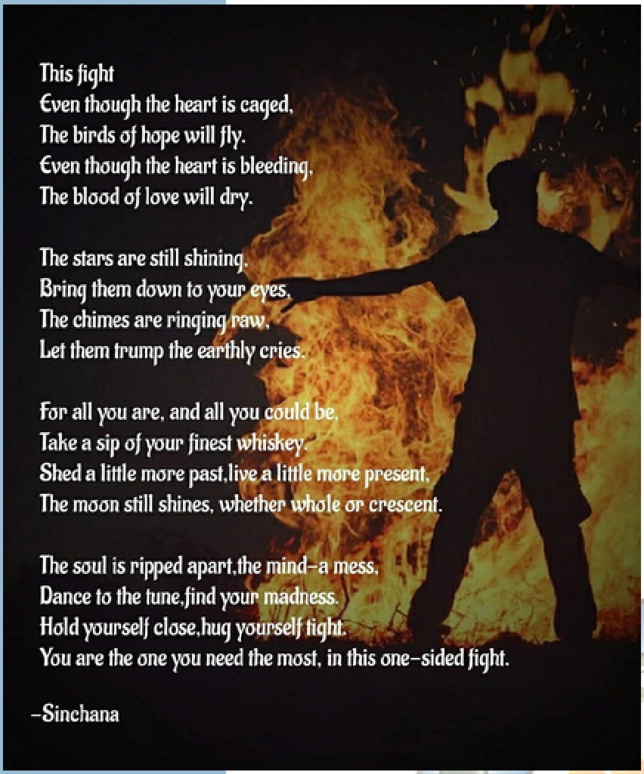
FROM EMERALD GREEN TO SAPPHIRE BLUE,
THOSE STROKES PAINT A MAJESTIC VIEW,
RIBBONS OF GOLD TWISTED ACROSS THE SKY,
THE CONSTELLATIONS HUM A SONG BY AND BY

-M SHANON AKANSHA,6B

JUST FOR FUN

- THERE ARE 10 TYPES OF PEOPLE IN THIS WORLD, ONE WHO CAN UNDERSTAND BINARY AND OTHER WHO DON'T.
- ROSES ARE RED VIOLETS ARE BLUE UNEXPECTED { ON LINE 52
- MISTAKES AREN'T MISTAKES UNLESS YOU KNOW CONTROL+Z

-CHAITRA R,6A



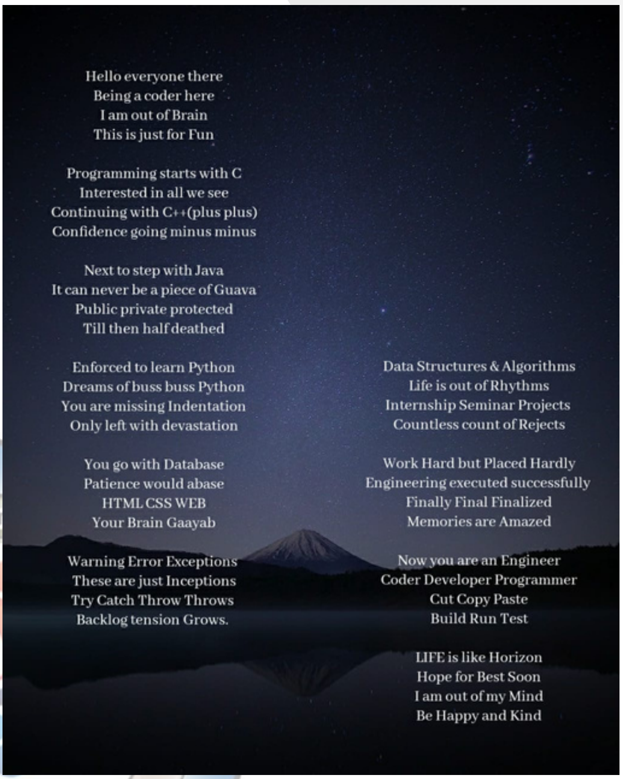
This fight
Even though the heart is caged,
The birds of hope will fly.
Even though the heart is bleeding,
The blood of love will dry.

The stars are still shining,
Bring them down to your eyes,
The chimes are ringing raw,
Let them trump the earthly cries.

For all you are, and all you could be,
Take a sip of your finest whiskey.
Shed a little more past,live a little more present,
The moon still shines, whether whole or crescent.

The soul is ripped apart,the mind-a mess,
Dance to the tune,find your madness.
Hold yourself close,hug yourself tight.
You are the one you need the most, in this one-sided fight.

-Sinchana



Hello everyone there
Being a coder here
I am out of Brain
This is just for Fun

Programming starts with C
Interested in all we see
Continuing with C++(plus plus)
Confidence going minus minus

Next to step with Java
It can never be a piece of Guava
Public private protected
Till then half deathed

Enforced to learn Python
Dreams of buss buss Python
You are missing Indentation
Only left with devastation

You go with Database
Patience would abase
HTML CSS WEB
Your Brain Gaayab

Warning Error Exceptions
These are just Inceptions
Try Catch Throw Throws
Backlog tension Grows.

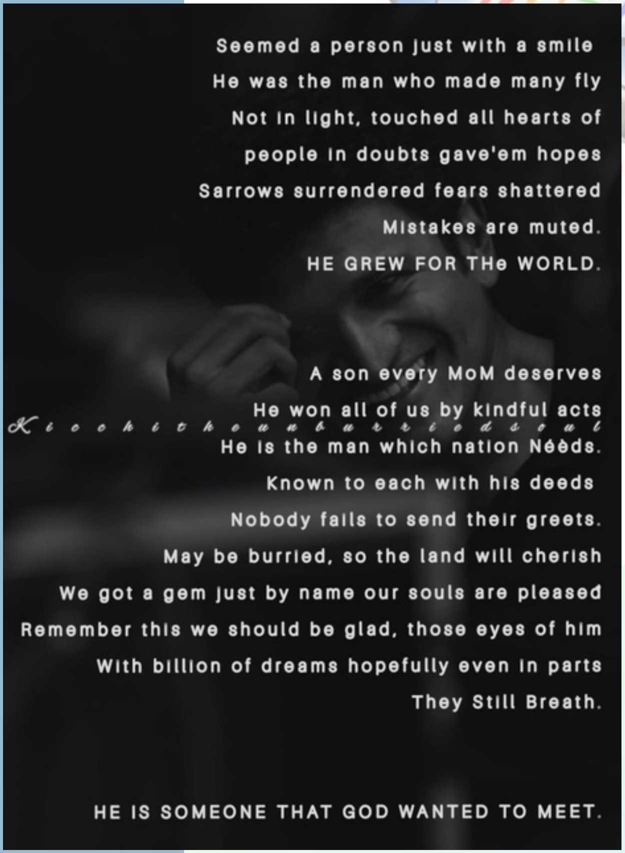
Data Structures & Algorithms
Life is out of Rhythms
Internship Seminar Projects
Countless count of Rejects

Work Hard but Placed Hardly
Engineering executed successfully
Finally Final Finalized
Memories are Amazed

Now you are an Engineer
Coder Developer Programmer
Cut Copy Paste
Build Run Test

LIFE is like Horizon
Hope for Best Soon
I am out of my Mind
Be Happy and Kind

_POOJA B,6A

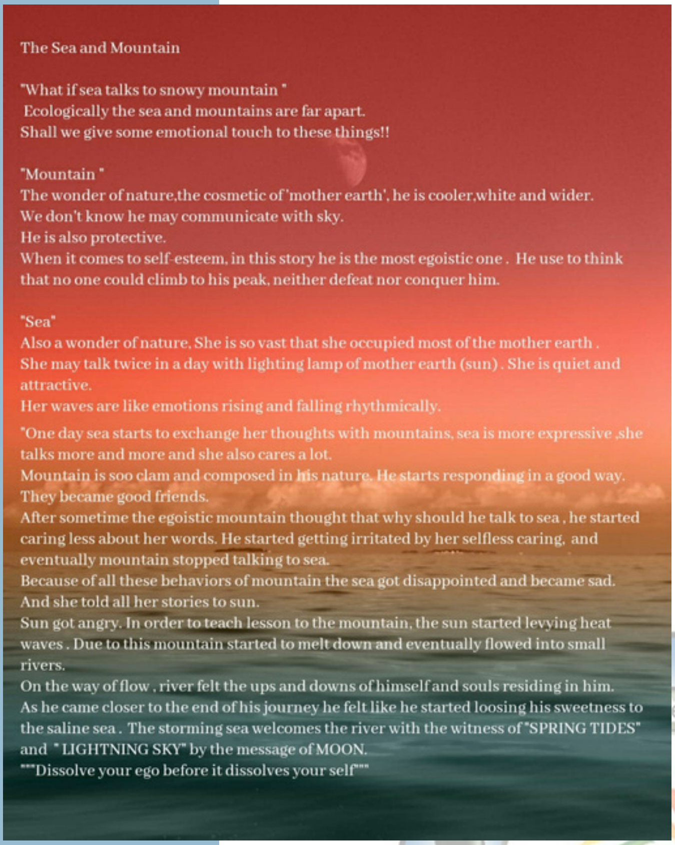


Seemed a person just with a smile
He was the man who made many fly
Not in light, touched all hearts of
people in doubts gave'em hopes
Sarrows surrendered fears shattered
Mistakes are muted.
HE GREW FOR THe WORLD.

A son every MoM deserves
He won all of us by kindful acts
K i c c h i t h e u n b u r r i e d s o u l
He is the man which nation Nėėds.
Known to each with his deeds
Nobody fails to send their greets.
May be burried, so the land will cherish
We got a gem just by name our souls are pleased
Remember this we should be glad, those eyes of him
With billion of dreams hopefully even in parts
They Still Breath.

HE IS SOMEONE THAT GOD WANTED TO MEET.

-SHRAVYA D GOWDA ,6B



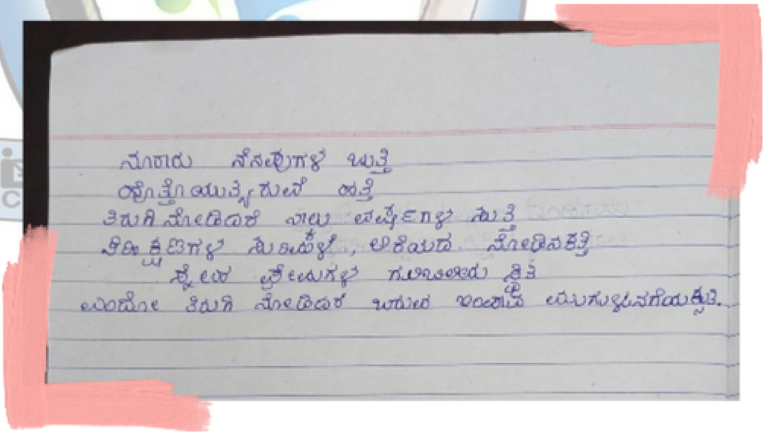
The Sea and Mountain

"What if sea talks to snowy mountain "
Ecologically the sea and mountains are far apart.
Shall we give some emotional touch to these things!!

"Mountain "
The wonder of nature,the cosmetic of 'mother earth', he is cooler,white and wider.
We don't know he may communicate with sky.
He is also protective.
When it comes to self-esteem, in this story he is the most egoistic one . He use to think that no one could climb to his peak, neither defeat nor conquer him.

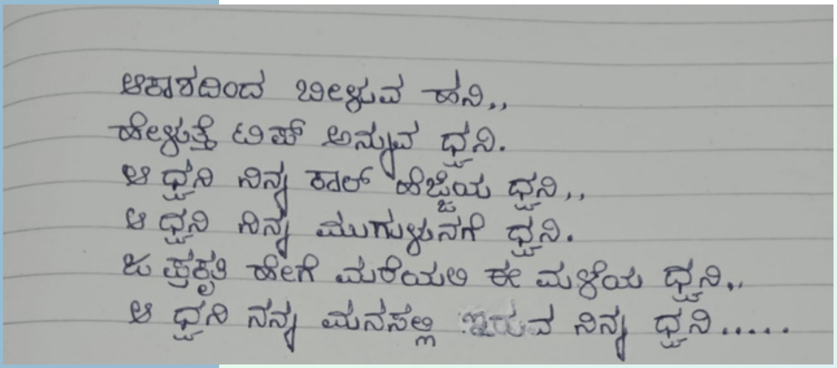
"Sea"
Also a wonder of nature. She is so vast that she occupied most of the mother earth .
She may talk twice in a day with lighting lamp of mother earth (sun) . She is quiet and attractive.
Her waves are like emotions rising and falling rhythmically.
"One day sea starts to exchange her thoughts with mountains, sea is more expressive ,she talks more and more and she also cares a lot.
Mountain is soo clam and composed in his nature. He starts responding in a good way.
They became good friends.
After sometime the egoistic mountain thought that why should he talk to sea , he started caring less about her words. He started getting irritated by her selfless caring, and eventually mountain stopped talking to sea.
Because of all these behaviors of mountain the sea got disappointed and became sad.
And she told all her stories to sun.
Sun got angry. In order to teach lesson to the mountain, the sun started levying heat waves . Due to this mountain started to melt down and eventually flowed into small rivers.
On the way of flow , river felt the ups and downs of himself and souls residing in him.
As he came closer to the end of his journey he felt like he started loosing his sweetness to the saline sea . The storming sea welcomes the river with the witness of "SPRING TIDES" and " LIGHTNING SKY" by the message of MOON.
""Dissolve your ego before it dissolves your self""

-NITHYA K G,6A



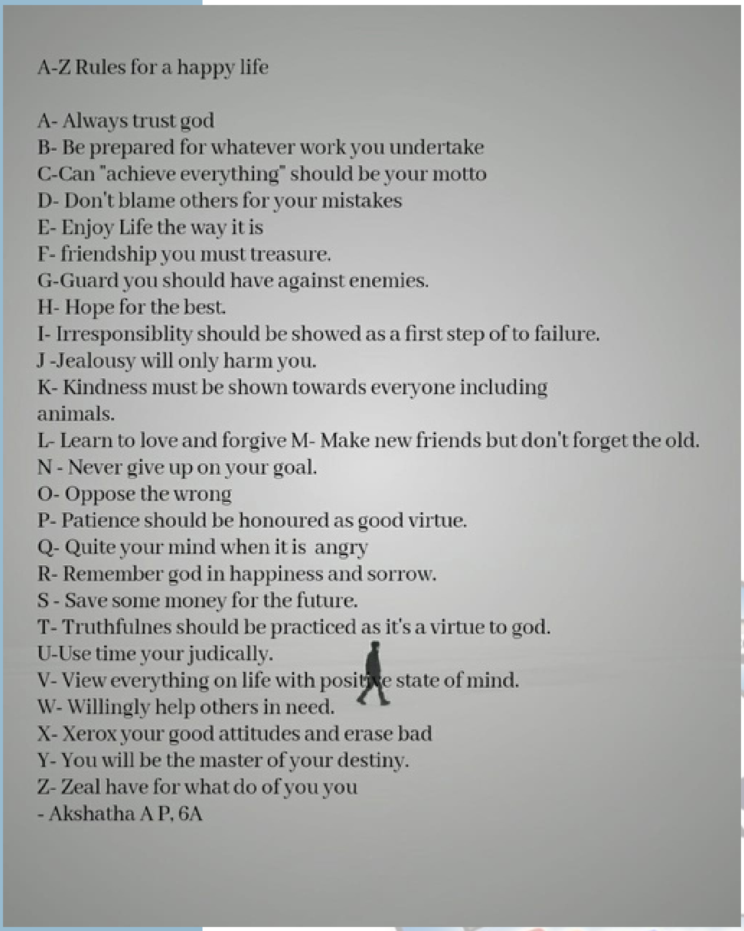
ನೂರಾರು ನೆನಪುಗಳ ಬುತ್ತಿ
ಹೊತ್ತೊಯ್ಯುತ್ತಿರುವೆ ಎತ್ತ
ತಿರುಗಿ ನೋಡಿದರೆ ಎಲ್ಲ ಅಷ್ಟೇನಾ ಸುತ್ತ
ತಿಳಿ ಕ್ಷಣಗಳ ಸುಂದರ್ಯ, ಅರಿಯದ ನೋವಿನಕತ್ತೆ
ಸ್ನೇಹದ ಪ್ರೇಮಗಳ ಗುಣಾವಳಿ ಚಿತ್ತ
ಎಂದೋ ತಿರುಗಿ ನೋಡಿದರೆ ಬರೀ ಒಂಚೂರು ಮುಗುಳ್ನಗೆಯಷ್ಟೆ.

-SYED AQEEL AHMED,6B



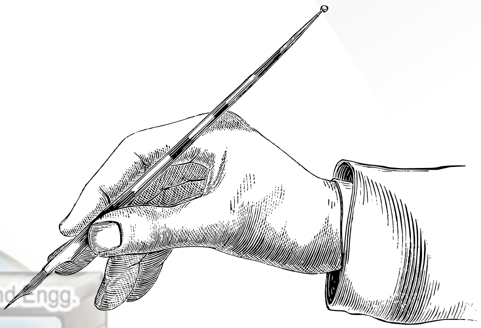
ಆಕಾಶದಿಂದ ಬೀಳುವ ಹನಿ,,
ಹೇಳುತ್ತೆ ಟಿಪ್ ಅನ್ನುವ ಧ್ವನಿ.
ಆ ಧ್ವನಿ ನಿನ್ನ ಕಾಲ್ ಹೆಜ್ಜೆಯ ಧ್ವನಿ,,
ಆ ಧ್ವನಿ ನಿನ್ನ ಮುಗುಳುನಗೆ ಧ್ವನಿ.
ಈ ಪ್ರಕೃತಿ ಹೇಗೆ ಮರೆಯಲಿ ಈ ಮಳೆಯ ಧ್ವನಿ,,
ಆ ಧ್ವನಿ ನನ್ನ ಮನಸಲ್ಲಿ ಇರುವ ನಿನ್ನ ಧ್ವನಿ....

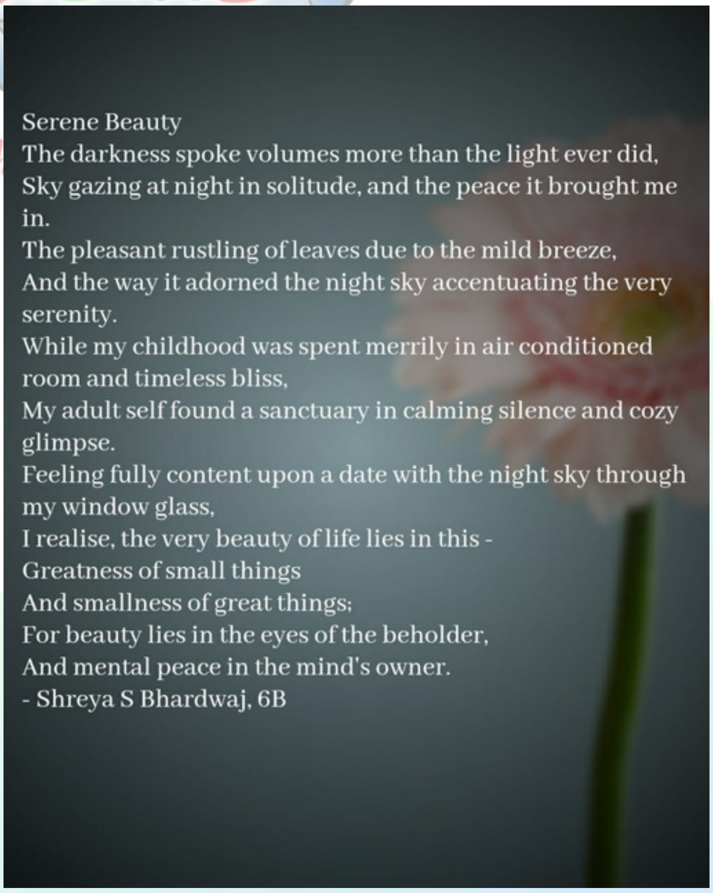
-RUKMINI S,6B



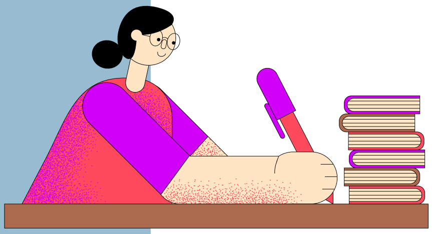
A-Z Rules for a happy life

A- Always trust god
B- Be prepared for whatever work you undertake
C-Can "achieve everything" should be your motto
D- Don't blame others for your mistakes
E- Enjoy Life the way it is
F- friendship you must treasure.
G-Guard you should have against enemies.
H- Hope for the best.
I- Irresponsiblity should be showed as a first step of to failure.
J -Jealousy will only harm you.
K- Kindness must be shown towards everyone including animals.
L- Learn to love and forgive M- Make new friends but don't forget the old.
N - Never give up on your goal.
O- Oppose the wrong
P- Patience should be honoured as good virtue.
Q- Quite your mind when it is angry
R- Remember god in happiness and sorrow.
S - Save some money for the future.
T- Truthfulnes should be practiced as it's a virtue to god.
U-Use time your judically.
V- View everything on life with positive state of mind.
W- Willingly help others in need.
X- Xerox your good attitudes and erase bad
Y- You will be the master of your destiny.
Z- Zeal have for what do of you you
- Akshatha A P, 6A





Serene Beauty
The darkness spoke volumes more than the light ever did,
Sky gazing at night in solitude, and the peace it brought me in.
The pleasant rustling of leaves due to the mild breeze,
And the way it adorned the night sky accentuating the very serenity.
While my childhood was spent merrily in air conditioned room and timeless bliss,
My adult self found a sanctuary in calming silence and cozy glimpse.
Feeling fully content upon a date with the night sky through my window glass,
I realise, the very beauty of life lies in this -
Greatness of small things
And smallness of great things;
For beauty lies in the eyes of the beholder,
And mental peace in the mind's owner.
- Shreya S Bhardwaj, 6B



TECH INSIGHTS: DEMYSTIFYING COMPLEX CONCEPTS

TITLE: HARNESSING THE POWER OF SANSKRIT: EXPLORING SANSKRIT AS A PROGRAMMING LANGUAGE

Introduction:
Sanskrit, the ancient language of india, is known for its precision, rich vocabulary, and grammatical structure. While it has predominantly been used in the realms of literature, philosophy, and spirituality, there is an emerging interest in utilizing sanskrit as a programming language. This article delves into the potential of sanskrit as a programming language, exploring its unique features and providing insights into its practical applications.

1. The roots of sanskrit as a programming language:
Sanskrit's grammatical structure aligns remarkably well with the syntax and rules of modern Programming languages. Its precise and logical nature makes it an intriguing candidate for coding Purposes. By leveraging the linguistic strengths of sanskrit, programmers can potentially develop Robust and expressive software systems.

2. Simplicity and readability:
Sanskrit's straightforward grammar and unambiguous syntax contribute to the simplicity and Readability of code written in this language. Its well-defined rules and consistent structure enhance Code maintainability and reduce the chances of errors. The use of sanskrit as a programming Language could foster codebases that are easy to understand, debug, and collaborate on.

3. Expressive power:
Sanskrit's extensive vocabulary allows programmers to articulate complex ideas and concepts with Conciseness and clarity. The language's wide range of linguistic features, such as compound words, Verb conjugations, and noun declensions, can be harnessed to create expressive code that captures The intricacies of algorithms and problem-solving approaches.

4. Integration with Indian Heritage and Culture:
By utilizing Sanskrit as a programming language, developers can bridge the gap between technology and India's rich cultural heritage. This approach fosters a sense of connection and promotes the preservation and dissemination of an ancient language in the modern digital world.

5. Practical Applications:
a. Natural Language Processing (NLP): Sanskrit's grammatical structure can enhance the accuracy and efficiency of natural language processing algorithms. NLP systems can leverage Sanskrit's rules to perform more nuanced semantic analysis, improve machine translation, and refine speech recognition capabilities.

b. Artificial Intelligence (AI) and Machine Learning (ML): Sanskrit's expressiveness and precision can be leveraged to develop sophisticated AI and ML algorithms. The language's ability to articulate complex mathematical and logical constructs can contribute to the development of advanced models and algorithms.
(Source: [2] - "Sanskrit for Artificial Intelligence")

c. Computational Linguistics: Sanskrit's well-defined grammar and extensive lexicon can aid in the development of computational linguistics applications. By incorporating Sanskrit into language processing systems, researchers can improve text analysis, semantic understanding, and machine comprehension.
(Source: [3] - "Computational Linguistics using Sanskrit")

Conclusion:

The utilization of Sanskrit as a programming language opens up exciting possibilities for software development. With its inherent precision, expressive power, and integration with India's cultural heritage, Sanskrit can contribute to advancements in various fields, such as NLP, AI, and computational linguistics. Exploring Sanskrit as a programming language is not only a technical endeavor but also an opportunity to preserve and promote the linguistic and cultural wealth of ancient India.

-NANDISH D, 6 A

TITLE: PĀṆINI'S AṢṬĀDHYĀYĪ

Pāṇini's Aṣṭādhyāyī, often referred to as the "Aṣṭādhyāyī," is an ancient Sanskrit treatise on grammar and linguistics. Written by the Indian grammarian Pāṇini in the 4th century BCE, it is considered one of the most influential works in the field of linguistics. The Aṣṭādhyāyī lays down comprehensive rules and principles for the structure, formation, and interpretation of Sanskrit sentences. The insights and techniques found in this seminal work can be valuable in building Sanskrit as a programming language. Here are some aspects of the Aṣṭādhyāyī that can be helpful:

1. Sūtras and Formal Rules: The Aṣṭādhyāyī is composed of a series of concise rules, known as sūtras, which serve as the foundation for generating and understanding Sanskrit sentences. These sūtras exhibit a remarkable level of brevity and encapsulate complex linguistic principles. Similarly, in programming languages, concise and formal rules can contribute to the clarity and efficiency of code.

2. Hierarchical Structure: Pāṇini's work follows a hierarchical structure, organizing linguistic elements into a system of rules, classes, and categories. This systematic approach allows for a logical organization of linguistic constructs and can inspire the development of hierarchical structures in the syntax and semantics of Sanskrit-based programming languages.

3. Generative Grammar: Pāṇini's approach to grammar is generative in nature, as he provides rules for the generation of valid sentences and for the analysis of complex linguistic structures. Similarly, programming languages utilize generative grammars to define the syntax and semantics of their constructs, enabling the generation and interpretation of code.

4. Formalism and Precision: The Aṣṭādhyāyī employs a formal and precise framework to describe the Sanskrit language. It defines rules for morphological and syntactical analysis, allowing for unambiguous interpretation of sentences. This emphasis on precision and formalism can be adapted to design programming languages that prioritize clarity, predictability, and robustness.

5. Computational Efficiency: Pāṇini's grammatical rules aim to capture the most economical and efficient way to express linguistic elements. This consideration of computational efficiency can be applied to programming language design, ensuring that code execution is optimized and that computational resources are utilized effectively.

6. Rule-based Systems: The Aṣṭādhyāyī provides a vast set of rules that govern the formation and interpretation of Sanskrit sentences. These rules can serve as a foundation for developing rule-based systems in programming languages, allowing for the specification of complex behaviors and logical constraints.

Conclusion

By studying and extracting insights from Pāṇini's Aṣṭādhyāyī, developers interested in building Sanskrit as a programming language can draw inspiration from its formalism, precision, hierarchical structure, generative grammar, and computational efficiency. Incorporating these principles can contribute to the development of a powerful and expressive programming language that embraces the inherent strengths of Sanskrit.

-NANDISH D, 6 A

TITLE: THE FUTURE OF VIRTUAL REALITY

Introduction:

Virtual Reality (VR) has made significant strides since its inception, transforming from a niche technology into a mainstream phenomenon. With the advent of powerful computing devices and advancements in graphics, VR has gained traction across various industries, from entertainment and gaming to healthcare and education. As we peer into the future, it becomes clear that virtual reality is poised to revolutionize the way we interact, learn, and experience the world around us.

1. Immersive Entertainment: In the future, virtual reality will revolutionize the entertainment industry, taking us beyond the confines of traditional screens. Imagine being fully immersed in a virtual world, where you can explore fantastical realms, interact with lifelike characters, and engage in immersive storytelling. Moreover, VR will extend beyond gaming to encompass virtual concerts, movies, and sporting events, allowing people to be present in the virtual arena and connect with others from around the globe

2. Enhanced Education and Training: Virtual reality holds tremendous potential for education and training. Imagine students being able to step into historical events, explore the human body from within, or travel to distant planets for science lessons. VR will provide a dynamic and engaging platform for immersive learning, allowing students to visualize complex concepts and interact with virtual objects and environments. Similarly, VR will revolutionize training programs by creating realistic simulations for industries such as healthcare, aviation, and military, enabling trainees to gain hands-on experience in a safe and controlled environment.

3. Virtual Collaboration and Telepresence: The future of work will see a paradigm shift with the integration of virtual reality. Virtual collaboration platforms will enable teams spread across the globe to work together in shared virtual spaces, fostering creativity, productivity, and a sense of presence. Meetings will no longer be confined to video calls; instead, colleagues will interact as avatars, sharing and manipulating 3D models and data. Additionally, VR will transform remote work by providing a sense of physical presence, reducing isolation, and enhancing communication.

4. Healthcare and Therapy: Virtual reality has the potential to revolutionize healthcare by enhancing diagnostics, treatment, and therapy. Surgeons can practice complex procedures in a virtual environment, reducing the risk to patients. VR can also aid in pain management by immersing patients in relaxing or distracting virtual environments. Furthermore, therapists can use VR to treat various mental health conditions, such as phobias, PTSD, and anxiety disorders, by creating controlled and immersive exposure therapy scenarios.

5. Social Impact and Empathy: Virtual reality has the power to evoke empathy and create social impact. By enabling users to step into someone else's shoes and experience different perspectives, VR can foster empathy and understanding for diverse cultures, social issues, and marginalized communities. VR experiences can raise awareness about global challenges, such as climate change and poverty, and inspire individuals to take action.

Conclusion:

The future of virtual reality holds immense potential to reshape our lives and unlock new dimensions of experience. From entertainment and education to healthcare and social impact, VR will permeate various aspects of our lives, offering immersive and transformative experiences. As technology continues to advance, we can look forward to a future where virtual reality becomes an integral part of our everyday lives, enriching our experiences and connecting us in ways we never thought possible. The journey has just begun, and the possibilities are limitless

-HAJIRA KOUSER 6TH A

TITLE: WHAT BRAIN-COMPUTER INTERFACE COULD MEAN FOR THE FUTURE WORK

Brain Computer Interfaces for Improving the Quality of Life of Older Adults and Elderly Patients Imagine if you could prepare your next presentation using only your thoughts. These scenarios might soon become a reality thanks to the development of braincomputer interfaces (BCIs). To put it in the simplest terms, think of a BCI as a bridge between your brain and an external device. As of today, we mostly rely on electroencephalography (EEG) — a collection of methods for monitoring the electrical activity of the brain — to do this. But, that's changing. By leveraging multiple sensors and complex algorithms, it's now becoming possible to analyze brain signals and extract relevant brain patterns.

Brain activity can then be recorded by a non-invasive device — no surgical intervention needed. In fact, the majority of existing and mainstream BCIs are noninvasive, such as wearable headbands and earbuds. The development of BCI technology was initially focused on helping paralyzed people control assistive devices using their thoughts. But new use cases are being identified all the time. For example, BCIs can now be used as a neurofeedback training tool to improve cognitive performance. A Toronto-based startup called "Muse" has developed a sensing headband that gives real-time information about what's going on in your brain. Researchers are also experimenting with "passthoughts" as an alternative to passwords. Soon, we might log into our various devices and platforms using our thoughts. When we perform mental tasks like picturing a shape or singing a song in our heads, our brains generate unique neuronal electrical signals.

A billion people could mentally hum the same song and no two brain-wave patterns generated by that task would be alike. An electroencephalograph (EEG) would read those brain waves using noninvasive electrodes that record the signals. The unique patterns can be used like a password or biometric identification.

-BHAVANA R M,6A

TITLE: IS TIME TRAVEL POSSIBLE?

Although many people are fascinated by the idea of changing the past or seeing the future before it's due, no person has ever demonstrated the kind of back-and-forth time travel seen in science fiction or proposed a method of sending a person through significant periods of time that wouldn't destroy them on the way. And, as physicist Stephen Hawking pointed out in his book "Black Holes and Baby Universes" (Bantam, 1994), "The best evidence we have that time travel is not possible, and never will be, is that we have not been invaded by hordes of tourists from the future."

Dueling theories

In 1905, Albert Einstein published the first part of his relativity theory, known as special relativity. In it, space and time are malleable; measurements of both space and time depend on the relative speed of the person doing the measuring.

"It is absolutely provable in special relativity that the astronaut who makes the journey, if they travel at very nearly the speed of light, will be much younger than their twin when they come back," says Janna Levin, a physicist at Barnard College in New York. Interestingly, time appears to pass just as it always does for both twins; it's only when they're reunited that the difference reveals itself Then, in 1915, Einstein came up with the second part of his theory, known as general relativity. General relativity renders gravity in a new light. With general relativity, things really start to get interesting. In this theory, a massive object warps or distorts space and time. Perhaps you've seen diagrams or videos comparing this to the way a ball distorts a rubber sheet. One result is that, just as travelling at a high speed affects the rate at which time passes, simply being near a really heavy object—like a black hole—will affect one's experience of time.

Can we use time travel in everyday life?

We can't use a time machine to travel hundreds of years into the past or future. That kind of time travel only happens in books and movies. But the math of time travel does affect the things we use every day. GPS satellites orbit around Earth very quickly at about 8,700 miles (14,000 kilometers) per hour. This slows down GPS satellite clocks by a small fraction of a second. However, the satellites are also orbiting Earth about 12,550 miles (20,200 km) above the surface. This actually speeds up GPS satellite clocks by a slighter larger fraction of a second.

-BHAVAN RM,6A

TITLE: POWER OF POSITIVE THINKING

For living a wonderful life positive attitude is very much important ,as we face various different situations in life being calm, being motivated ,leading joyful and cheerful life positive attitude is important. Positive thinking doesn't mean that you ignore life's less pleasant situations. Positive thinking just means that you approach unpleasantness in a more positive and productive way. You think the best is going to happen, not the worst. You can learn to turn negative thinking into positive thinking. The process is simple, but it does take time and practice — you're creating a new habit, after all. Following are some ways to think and behave in a more positive and optimistic way: • Identify areas to change. If you want to become more optimistic and engage in more positive thinking, first identify areas of your life that you usually think negatively about, whether it's work, your daily commute, life changes or a relationship.

You can start small by focusing on one area to approach in a more positive way. Think of a positive thought to manage your stress instead of a negative one.

• Check yourself. Periodically during the day, stop and evaluate what you're thinking. If you find that your thoughts are mainly negative, try to find a way to put a positive spin on them.
• Be open to humor. Give yourself permission to smile or laugh, especially during difficult times. Seek humor in everyday happenings. When you can laugh at life, you feel less stressed.

• Follow a healthy lifestyle. Aim to exercise for about 30 minutes on most days of the week. You can also break it up into 5- or 10-minute chunks of time during the day. Exercise can positively affect mood and reduce stress.
• Surround yourself with positive people. Make sure those in your life are positive, supportive people you can depend on to give helpful advice and feedback. Negative people may increase your stress level and make you doubt your ability to manage stress in healthy ways.
• Practice positive self-talk. Start by following one simple rule: Don't say anything to yourself that you wouldn't say to anyone else. Be gentle and encouraging with yourself. If a negative thought enters your mind, evaluate it rationally and respond with affirmations of what is good about you. Think about things you're thankful for in your life.

Practicing positive thinking every day

If you tend to have a negative outlook, don't expect to become an optimist overnight. But with practice, eventually your self-talk will contain less self-criticism and more self-acceptance. You may also become less critical of the world around you. When your state of mind is generally optimistic, you're better able to handle everyday stress in a more constructive way. That ability may contribute to the widely observed health benefits of positive thinking. Along with positive thinking we have to concentrate on what we are seeing, what we are hearing and what we are taking because, there is saying that "THE MIND IS EVERY THING. WHAT YOU THINK YOU BECOME", what we see, hear and talk that we become, every act we are doing will be effecting our character, The way we are. Life is not so hard to live ,but if we no how to tackle the situation and by practicing the art of living we can absolutely lead the joyful and a wonderful life.

"CHANGE YOUR THINKING, CHANGE YOUR LIFE"

-LIKHITHA R 6TH A

Title: The Future-Proof Nature of Java: Building on a Strong Foundation

Introduction:

In the fast-paced world of technology, staying relevant and future-proofing programming languages is a top priority for developers and organizations. Java, a widely used and mature programming language, has consistently proven its ability to adapt, evolve, and remain future-proof. In this article, we will explore the reasons behind Java's enduring popularity, its strong foundations, and the measures taken by the Java community to ensure its continued relevance in the ever-changing technological landscape.

1. Platform Independence and Compatibility:
Java's "Write Once, Run Anywhere" principle has been one of its greatest strengths. Java programs are compiled into bytecode, which can run on any platform with a Java Virtual Machine (JVM). This cross-platform compatibility allows developers to write code once and deploy it across a wide range of devices, from desktop computers to smartphones, embedded systems, and the cloud. As technology continues to advance, Java's platform independence ensures its relevance in a world of diverse computing environments.

2. Continuous Evolution:
Java's evolution is driven by its strong community and the stewardship of Oracle, the organization responsible for its development. Regular updates and releases introduce new features, performance enhancements, and security improvements. Recent versions, such as Java 8, 9, 10, 11, and beyond, have brought significant enhancements, including lambda expressions, modularization (Project Jigsaw), and improved garbage collection. This commitment to continuous improvement ensures that Java remains up-to-date with emerging trends and technological advancements.

3. Robust Ecosystem and Libraries:
Java boasts a vast ecosystem of libraries, frameworks, and tools that empower developers to build sophisticated and scalable applications efficiently. The Java Development Kit (JDK) provides an extensive set of APIs for various tasks, including networking, database connectivity, user interfaces, and more. Popular frameworks like Spring, Hibernate, and JavaFX simplify application development and integrate seamlessly with modern architectural patterns. This rich ecosystem enables Java developers to leverage existing resources and accelerate the development process, irrespective of the industry they are working in.

4. Backward Compatibility:
One of Java's core principles is backward compatibility, ensuring that applications developed in earlier versions of Java continue to work seamlessly on newer versions. This commitment to maintaining backward compatibility is vital for businesses that rely on Java for critical systems. It allows organizations to adopt newer Java versions gradually, without the need for a complete rewrite of existing codebases, thereby reducing costs and minimizing disruption.

5. Industry Adoption and Job Market:
Java's widespread adoption across industries has contributed significantly to its future-proof nature. It is the backbone of numerous enterprise systems, banking applications, e-commerce platforms, and large-scale projects. The demand for Java developers remains consistently high, with an extensive job market offering diverse career opportunities. This popularity and industry support provide a solid foundation for Java's long-term viability.

Conclusion:

Java's future-proof nature stems from its platform independence, continuous evolution, robust ecosystem, backward compatibility, and widespread industry adoption. Its ability to adapt to changing technology trends and its thriving community ensure that Java will remain a prominent programming language for years to come. As developers embrace newer Java versions and harness its extensive libraries and frameworks, Java continues to demonstrate its strength in building scalable, reliable, and secure software solutions.

-JUNAIDH FARDEEN, 6A

Title: How to Fight Climate Change as a Software Engineer

Software has an impact on climate change and we as software engineers can make a difference. By keeping the created carbon emissions in mind and doing what is possible to reduce carbon emissions caused by software, we can contribute to the fight against climate change.
Waiting for data centers to fully run on renewable energy is not enough and will take too long. We need to decrease the amount of energy that software consumes, in addition to increasing the amount of renewable energy that powers the data centers in order to speed up this transition.
Huge amounts of energy are wasted every day by software blocking space and consuming energy at data centers without being used most of the time. We need to consequently scale software down to zero and remove unused deployments from data centers.
It is worth taking a look at the actual resource consumption of software; efforts to reduce this resource consumption pay off in terms of lower energy and hardware consumption. The impact looks small initially, but scaling effects turn it into significant numbers.
Take the carbon intensity into account when choosing a data center or public cloud region - the carbon emissions caused by a data center can vary a lot when running the exact same workload. Choosing a region with lower carbon intensity helps quite a bit to run your workload with less carbon emissions.
We need to reduce and eliminate greenhouse gas emissions in order to stop climate change. There is no way around this. But what is the role that software plays here? And what can we - as software engineers - do about this? Let's take a look under the hood to uncover the relationship between greenhouse gas emissions and software, learn about the impact that we can have, and identify concrete ways to reduce those emissions on a day-to-day basis when creating and running software.
Software is everywhere. We use software all the time. There are probably millions of lines of software running in your pocket, on your smartphone, all the time. There are millions of lines of software running on devices all around us, and there are trillions of lines of software running in data centers around the globe that we use every day, every hour. You can't make a phone call anymore without huge amounts of software being involved, you can't buy your groceries at the store or use your bank account without software being involved.

If you look behind the scenes of all this software, you will find huge amounts of greenhouse gas emissions - the driving factor of climate change - being produced and emitted to the atmosphere in this process, caused by a variety of activities around software. The hardware that is used to run the software needs to be produced, the data center that runs the software needs to be powered with energy, needs to be cooled, data needs to be transferred over the network, and so on. The more you look into the details of software, the more aspects you identify that cause greenhouse gas emissions - directly or indirectly.
As an example, we can look into data centers that run huge amounts of software every second. We know that the total energy consumption of data centers around the globe is significant - and will increase even further in the future. We are talking here about something in the range of maybe 10% of the energy produced on the entire planet being consumed by data centers in the near future. This is huge. And it is only one of many aspects here.
Energy is a key factor

Energy production is still a major driver of greenhouse gas emissions. Even if you hear slogans of "we use 100% renewable energy", this usually doesn't mean that your data center really runs on renewable energy all the time. It typically means that the provider buys (or produces) renewable energy in the same amount as the data center uses over a period of time.
Unfortunately the energy consumption of a data center doesn't align with the energy production from renewable sources all the time. Sometimes more renewable energy is being produced than consumed by the data center, but sometimes the opposite happens: the data center needs more energy than is currently available from renewable sources. In those situations, the data center depends on the energy grid to fill in the gaps. And consuming energy from the grid means to depend on the energy mix that is available on the grid at that moment. The exact mix heavily depends on the country, the location within the country, and the exact time. But in almost all cases this mix includes energy being produced from emitting CO2 into the atmosphere (largely from burning coal, gas, and oil).
The companies who operate large data centers try to avoid this situation, for example by locating the data centers in locations with cool weather conditions (like Finland), so that less energy is needed for cooling. Or they locate data centers close to renewable energy production sites like windparks or hydro-based power stations. But running a data center on renewable energy all the time is still a huge challenge. We will get there, but it will take a long time.
The good news is that we as software engineers can help to accelerate this transition.
What can we do?

There are basically four fundamental things that we as software engineers can keep an eye on to accelerate the transition to run all our software on 100% renewable energy all the time:
Delete workloads that are no longer used
Run workloads only when necessary
Move workloads to a low carbon location and time
Use fewer resources for your workload
Delete workloads that are no longer used

Sometimes we allocate resources at a data center for a certain workload, we deploy and run the workload, and then, we forget that this workload exists, that the workload silently continues to run, and blocks allocated resources from being used elsewhere. Studies have revealed that these so-called "zombies" are a real problem. Jonathan Koomey and Jon Taylor revealed in their analysis of real-world data centers (Zombie/Comatose Server Redux) that between 1/4 to 1/3 of all running workloads are zombies: they are completely unused and non-active, but they block allocated resources and therefore consume significant amounts of energy.
We need to clean up our data centers from these zombies. That alone could help reduce the energy consumption significantly. Unfortunately, we don't have the tools yet to automatically identify zombie workloads in data centers or on public clouds. Beyond the fact that this is a huge opportunity for new and innovative projects in this space, we need to help ourselves in the meantime and manually identify those zombie workloads

The simple first step is, of course, to manually walk through all the running workloads to see if we immediately see a workload that we forgot about and/or that doesn't need to run anymore. Sounds trivial? Maybe. But this usually surfaces surprisingly many zombie workloads already. So doing this little annual (or monthly, or weekly) stock-taking and removing those unused workloads already makes a difference..

In addition to that, we can use regular observability tools for this job and look at usage numbers. The number of HTTP requests or the monitoring of the CPU activity are good examples of metrics to manually look at for a period of time to see if a workload is really used or not.
Run workloads only when necessary

Another interesting outcome of the study mentioned above is that, beyond zombie workloads, there is a large amount of workloads that are not being used most of the time. Their usage is not at zero (like zombie workloads are), but at a very low frequency. The cohort that the study discussed were workloads that were active for less than 5% of the time. Interestingly, this cohort counted for roughly another 1/3 of all analysed workloads.
When looking at those workloads, we need to keep in mind that having those workloads deployed and running consumes energy 100% of the time. The amount of energy that non-active workloads consume is definitely less than the same workload being used at 100% (due to energy saving technologies being applied at the microprocessor level, for example), but the total energy consumption that is related to the workload is still significant (probably something around 50% of the energy consumption when running under load). The ultimate goal here is to shutdown those workloads entirely when they are not used.
This is something that software architects and software engineers need to take into account when designing and writing software. The software needs to be able to startup quickly, on-demand, and needs to be capable of running in many possibly very short cycles - instead of a more classical server architecture that was built for server applications running for a very long time.
The immediate example that comes to mind are serverless architectures, allowing microservices to startup fast and run only on demand. So this is nothing that we can easily apply to many existing workloads right away, but we can keep this option in mind when writing or designing new or refactoring existing software.
Move workloads to a low carbon location and time

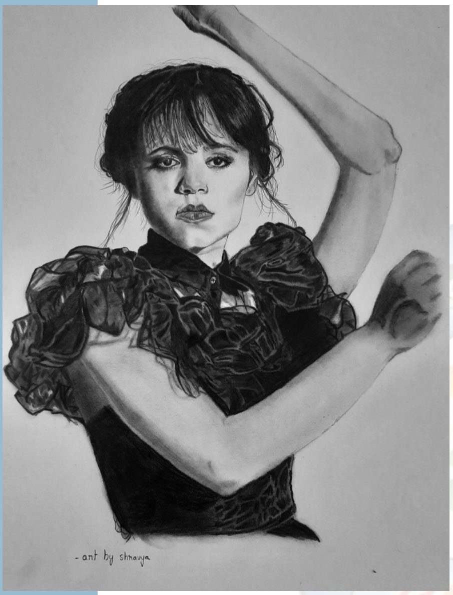
One of the challenges of powering data centers with renewable energy is the fact that renewable energy production is usually not at a constant level. The sun doesn't shine all the time and the wind doesn't blow all the time with the same intensity. This is one of the reasons why it is so hard to align the power consumption of data centers with the power produced from renewable sources. Whether the data center produces renewable energy on-site or consumes energy from the grid while purchasing green energy somewhere else doesn't really make a big difference with regards to this specific problem: each data center has different characteristics with regards to the energy mix it consumes during the day.
Fortunately, we can help this situation by moving workloads around in two dimensions: space and time. In case workloads need to run at a specific moment (or all the time), we can choose the data center with the best energy mix available. Some cloud providers already allow some insights into this, giving you an overview on the regions and their level of green energy. Others do not (yet), but you should ask for it. This is important data that should influence the decision of where to run workloads.
The second dimension here is time: renewable energy is not available at a constant level. There are times when more renewable energy is available and can power all the workloads, whereas there are other times when not enough green energy is around. If we can adjust the timing of when we run the software

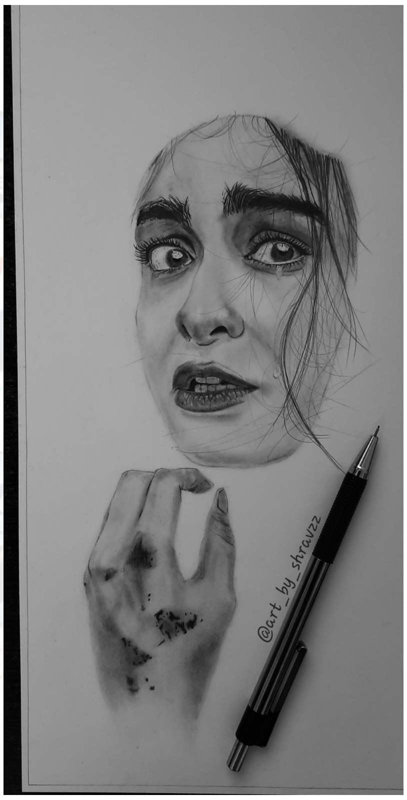
Keep this in mind when writing software and see if you can deploy your software in a way that allows the data center to move it around within certain boundaries or conditions. It helps data centers to adjust the load depending on the carbon intensity of the available power and therefore reduce carbon emissions.
Use fewer resources for your workloads

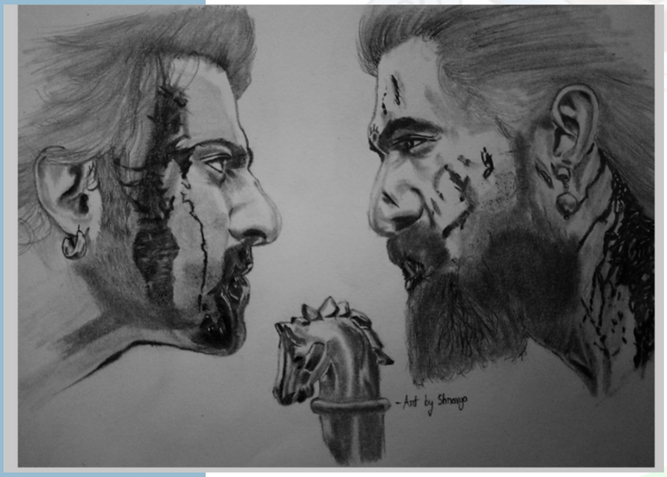
The last chapter of these various efforts is to use as few resources as possible when running the software. The rule of thumb for software engineers that I found during my studies for this purpose is to “try to run your software with less hardware.” Most of the other, more detailed suggestions and guiding principles can be derived from this simple rule of thumb.
Let’s assume you run your software in a containerized environment like Kubernetes. When running the workload, you define the resource requirements for your workload, so that kubernetes can find a place on a node of your cluster that has enough free space to schedule your workload within the constraints you defined. Whether your software actually uses those defined resources or not doesn’t really matter that much. The resources are reserved for your workload. They consume energy - even if those resources are not used by your workload. Reducing the resource requirements of your workload means to consume less energy and might even lead to more workloads being able to run on the node, which - in the end - even means to have lower hardware requirements for your cluster in total - and therefore less carbon emissions from hardware production, hardware upgrades, cooling of the machines, and powering them with energy.
Sometimes talking about using fewer resources for a workload sounds like talking about tiny little bits and pieces that don’t change the game, that don’t move the needle in the overall picture. But that is not true.
If we talk about small wattage numbers for memory or CPUs running in idle mode, those numbers sum up pretty quickly. Think about how easy it is to scale your software. You can scale it up to multiple, maybe hundreds or even thousands of instances running in the cloud. Your wattage numbers increase in the same way. Don’t forget that. When we talk about saving 100 Watts of CPU consumption for your application because you can deploy it on an instance with only four cores instead of six, it sounds small. But when we scale this application to 100 instances, it means saving 100 Watts per instance * 100 instances = 10000 Watts. Suddenly that is a lot. If we do this for every application that we run in the cloud or our own data center, energy consumption gets reduced quite a bit.
But we need to change our mindset for this. Sometimes we find ourselves thinking in the opposite direction: “Let's better give the application a bit more memory to make sure everything goes fine, to make sure we have a buffer, just in case…” We need to rethink that and change our perspective into the opposite direction. The question in our mind should be: “Can we run this application with less memory?”, or “Can we run this application with less CPU?”, or both.
Defining and running realistic load tests in an automated way can help here. The environments for those load tests can be defined with the new perspective in mind by reducing the available resources step by step. Watching the resource consumption using regular profiling and observability tools can surface the necessary data to find out when and why resource limits are hit - and where we need to optimise the software to consume less.Unfortunately, we don’t have all the tools yet to directly observe and measure the energy consumption of individual workloads or the carbon emissions caused by the consumed energy.

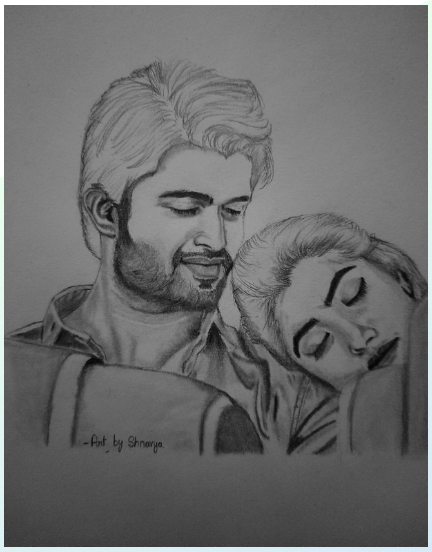
-SURFING RIAZ KHAN, 6A

PALETTE OF IMAGINATION: JOURNEY INTO THE ARTISTIC REALM

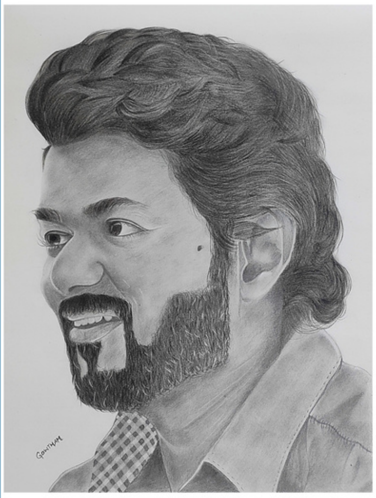








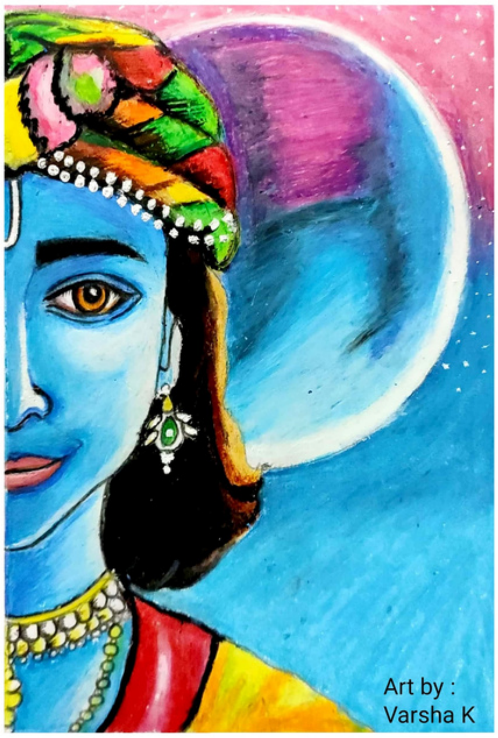
- SHARANYA

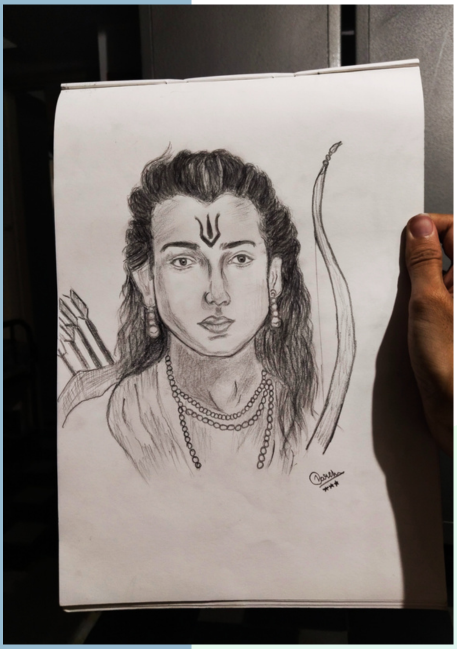


-GOUTHAM

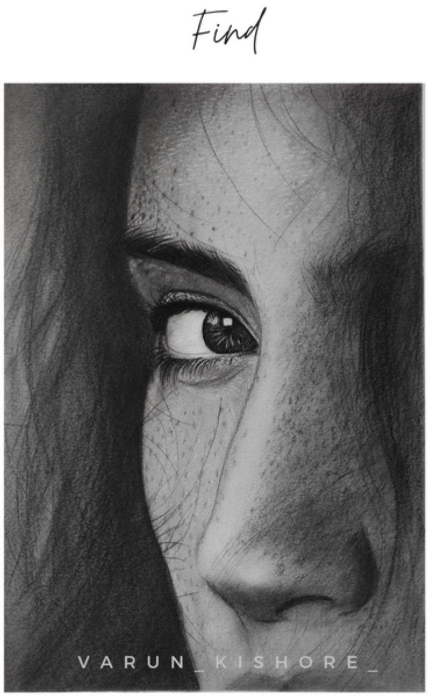
-DHRUTHI

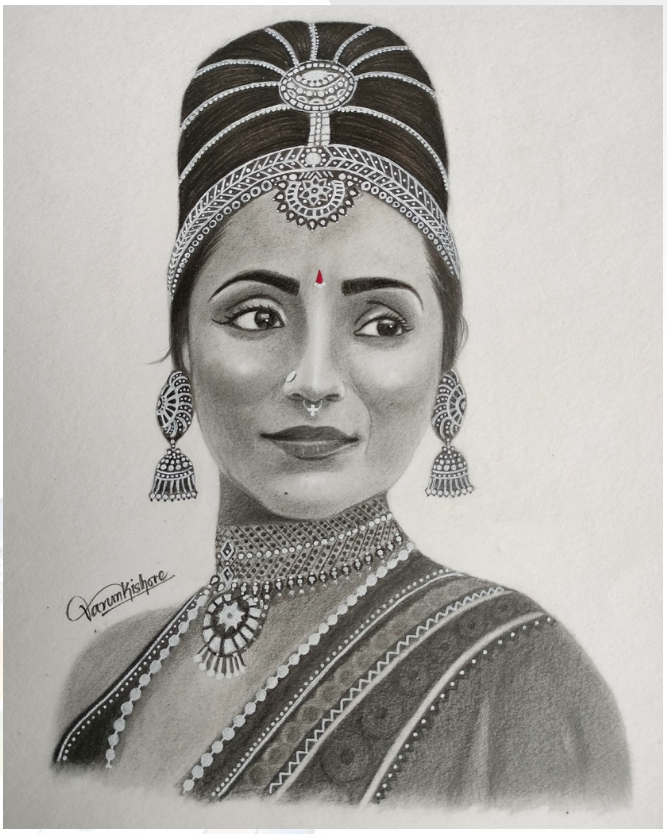
-LIKITHA R





-VARSHINI J M





STAEDTLER Mars Lumograph
VarunKishore

WORK BY VARUN KISHORE

PICTORIAL TIME CAPSULES: REFLECTING ON PAST EVENT CELEBRATIONS

PAINTING BY MYTHRI

INAUGARATION OF COSMOS 22-23

INAUGARATION OF ANVESHANA

VALEDICTORY OF ANVESHANA

STUDENT DEVELOPMENT PROGRAM ON JAVA BY DR. JALESH KUMAR

EXPERT TALK ON PERSONALITY DEVELOPMENT BY RITESH BHAT