

ASCII NEWSLETTER

ASSOCIATION OF STUDENTS OF COMPUTER SCIENCE FOR
INFORMATION INTERCHANGE

WHAT'S INSIDE THIS ISSUE

Let's know about FOSS!

PAGE 1

Your guide to the world
of Linux – 1

PAGE 2

Open Source Projects

PAGE 11

Workshop on 'Recent
trends in databases'

PAGE 14

Brain Teaser!

PAGE 16

Student Achievements

PAGE 17

Art Corner

PAGE 19

Editorial Team

PAGE 20



With warm greetings we welcome our viewers to the Issue 2 of our ASCII Newsletter. Through this issue we shall take you on a journey through the world of FOSS. It is our endeavour to provide maximum possible details on various Open Sources Technologies and projects based on Web Development and Designing. These projects will throw light on the intricacies of web development. We hope our projects will be of great interest to our readers. We have also included students' achievements in this newsletter!

Department of Computer Science and Engineering

Vision

To be acclaimed internationally for excellence in teaching and research in Computer Science & Engineering, and in fostering a culture of creativity and innovation to responsibly harness state-of-the-art technologies for societal needs.

Mission

Mission 1: To assist students in developing a strong foundation in Computer Science and Engineering by providing analytical, computational thinking and problem solving skills.

Mission 2: To inculcate entrepreneurial skills to develop solutions and products for interdisciplinary problems by cultivating curiosity, team spirit and spirit of innovation.

Mission 3: To provide opportunities for students to acquire knowledge of state-of-the-art in Computer Science and Engineering through industry internships, collaborative projects, and global exchange programmes with Institutions of international repute.

Mission 4: To develop life-long learning, ethics, moral values and spirit of service so as to contribute to the society through technology.

Mission 5: To be a premier research-intensive department by providing a stimulating environment for knowledge discovery and creation.

Programme Educational Objectives (PEOs)

The Computer Science & Engineering Program graduates will

PEO1: Strive on a global platform to pursue their professional career in Computer Science and Engineering.

PEO2: Contribute to product development as entrepreneurs in inter disciplinary fields of engineering and technology.

PEO3: Demonstrate high regard for professionalism, integrity and respect values in diverse culture, and have a concern for society and environment.

Programme Outcomes (PO's) and Programme Specific Outcomes (PSO's)

PO1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3: Design and development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6: The engineer and society: Apply reasoning informed by the contextual knowledge to Assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of and need for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PSO1: Adopt Standard Practices: Ability to design and engineer, innovative, optimal and elegant computing solutions to interdisciplinary problems using standard practices, tools and technologies.

PSO2: Research and Innovation: Ability to learn emerging computing paradigms for research and innovation

Free and Open-Source Software(FOSS)

- Lakshika K, CB.EN.U4CSE19663

Let's know what is a FOSS!

Free and Open Source Software (FOSS) is in contrast of proprietary software, where the software is hidden under restrictive copyright licensing and the source code is usually hidden from the users. To be more specific, it is like giving people free license to use, copy, study, and change the software in any way, encouraging them to come up with various designs. FOSS allows the user to inspect the source code and provides a high level control of the software's functions compared to proprietary software.

Freedoms that a FOSS provides to its user:

- Freedom 1: Allowing to change the source code.
- Freedom 2: Redistribute the exact source code.
- Freedom 3: Redistribute the altered source code.

Some beneficial advantages of FOSS!

- Decreased software costs.
 - Increased security and stability.
 - Protecting privacy, education, and giving users more control over their own hardware.
- In present day's scenario the use of Linux and descendants of BSD (Berkeley Software Design) are widely accessed, powering millions of servers, desktops, smartphones, and other devices.



How students can be benefited by FOSS?

One of the main advantage of FOSS is sharing of resources which includes knowledge, time, effort, code, money etc. It can be a platform for the student communities to showcase their work knowledge and skills in various fields. Leading IT organisations are using open source software for its competitive advantage and lower costs, where students can find great job opportunities to nurture their passion for programming and to make a difference. By working on a FOSS platform students can get good exposure to the real-world problems, learn from experts and seek their work experience, build new networks and have chances to get recruited by leading companies!!

Your guide to the world of Linux — 1

- Nirmal K , CB.EN.U4CSE19038

My Linux journey started a long time back, when the latest version of Ubuntu was 14.04 LTS and the world was yet to witness the pandemic. Or the movie Dunkirk. Or Brexit. Simpler times. Using my experience growing up with Linux, I've decided to come up with a series called 'Your guide to the world of Linux'. Let's get started!

What is Linux?

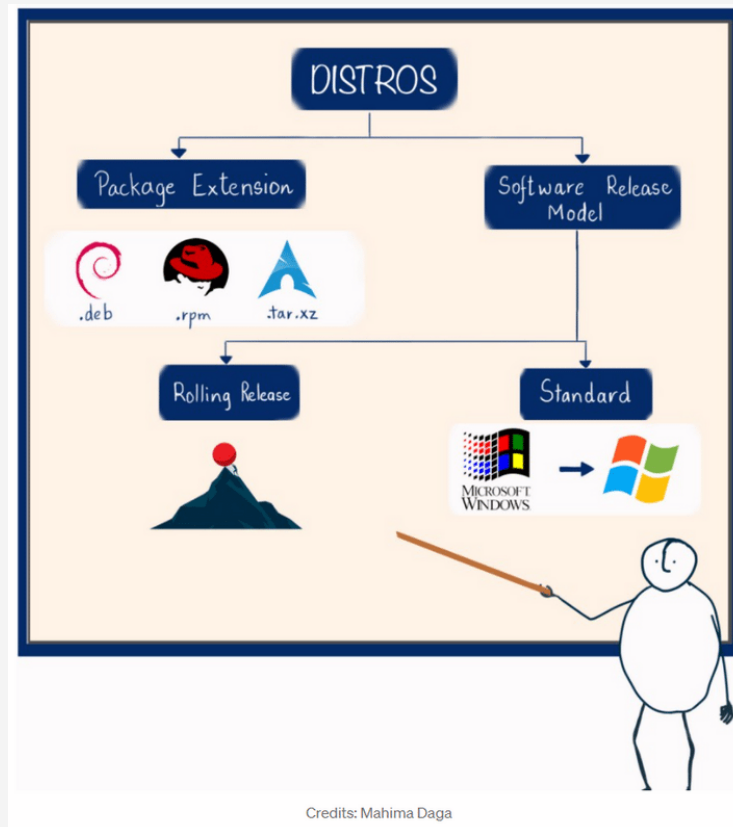
Contrary to popular belief, Linux is not an Operating System per se. The term 'Linux' itself refers to only the kernel of the operating system — the piece of software that talks to the hardware and allows other programs and users to make use of the computer. For this to be considered an actual Operating System, other components like a User Interface, system utilities and file systems have to be baked in.

This is what makes using Linux as your operating system a good idea. Right from the 'init' process to the User Interface, the user has choices at every single step. Contrast this with Operating Systems like Microsoft Windows and macOS which have software that's deemed the best by its developers and not much in the name of alternatives, and you can see why Linux is referred to as the tinkerer's dream Operating System.

Downside, this makes the world of Linux large, fragmented and confusing for a beginner. There are thousands of Operating Systems based on the Linux kernel, often differing in only very minute ways that it's difficult to evaluate and choose the best one to suit one's use case.

These thousands of Operating Systems are all called Linux distributions — distros for short. They're all different in some way, and they all offer something that most other operating systems can't in today's world. Choice.

The usual differentiator that I use to compare these Operating Systems are their software formats. Let's get into it!



Linux Software

Microsoft Windows users are used to the extension '.exe', and Android users to the extension '.apk'. In the Linux world, software is packaged by different distros into different file formats. And while there are ways to install packages built for one distro on a different distro, from the perspective of an average user this is going to be pointless. So the availability of a particular piece of software in a particular format might as well dictate the user's choice of Linux distro. So, what are the popular options?

- .deb
- .rpm
- .tar.zst
- .eopkg

These were developed to make installing software on Linux easier. Earlier, files were just simply compressed and left with instructions for the user to compile and then install — it was often very messy. With the advent of these packages though, distributions have been able to come up with package managers. They're much like the Play Store, App Store or the Windows Store, a collection of software packages that can be installed on the system, with options to install online and offline packages as well as options to uninstall packages.

The veterans on this list are the .deb and .rpm package formats. The former is used by Debian and its derivatives, which includes the popular Ubuntu and all its derivatives. The .rpm format is in use by Red Hat Enterprise Linux (RHEL) and its cousin Fedora, alongside SUSE and its cousin openSUSE (and CentOS, and all their derivatives. Phew!). The other package, .tar.xz, is used by Arch Linux and its derivatives. This the new, cool kid in the block, having been released in the year 2002, compared to 1993 for Debian and 1994 for Red Hat. And with Arch Linux, we introduce another important differentiator, the type of releases.

Software release model

To elaborate, there are two ways in which software is released. One is the rolling release model and the other is called the standard release model.

The standard release model is what most of us are used to, there are major releases after a period of time, with its own name and a large new feature set. This is seen in most desktop software today — we have Windows 3, 95, 98, ME, 2000, XP, Vista, 7, 8, 8.1 and 10 for example. In a similar way, most Linux distributions these days have version numbers — Ubuntu 20.04, Fedora 34, openSUSE LEAP 13.2, and so on.


The rolling release model on the other hand, is something different. There is no version numbering for the Operating System itself. Once installed, the user just keeps updating individual software packages to their latest versions. This way, users stay on the latest versions of every package and don't have to wait for specific versions to get new features.

Both models have their pros and cons, as is to be expected. With the standard release system, the new set of packages are tested extensively to make sure they work with each other and hence offer better stability. On the downside, features take a longer time to reach the end user.

In the rolling release cycle, software reaches the end user faster but the individual pieces of software might not always go well together. This might result in errors that might render the system unusable. It's a rare but possible occurrence and is often a deterrent for people wanting to try out rolling release distros.

As of now, we've come across two different differentiators — package format and software release model. These are the most important differentiators, the ones you're stuck with once you choose a distro.

Info time!



THE ART OF CHOOSING THE RIGHT LINUX DISTRO

Package format is one of the things to watch out for - do you wanna go for an OS that uses the **.deb**, **.rpm**, or **.tar.xz** format? Or something totally different?

Rolling release or standard release model? Immediate **updates** or staggered **upgrades**?

THE FORMAT WARS

Popular formats:

- **.deb** : Managed using the 'apt' package manager
- **.rpm** : Managed using different package managers - dnf, yum, zypper
- **.tar.xz**: Managed using the 'pacman' package manager

.DEB



Ubuntu



Debian



Kali Linux

.RPM



Red Hat



OpenSUSE



Fedora

.TAR.XZ



Arch Linux



Manjaro

HOW DO YOU WANT YOUR UPDATES SERVED?

Weeell there are two options:

- **Standard:** Versioned software releases. Windows, macOS, iOS, Android. You know what we're talking about. New versions with branding every now and then.
- **Rolling release:** There's no concept of version. If I publish an update to a piece of software I wrote, it's tested, approved and pushed to all end users. No waiting for a version upgrade to get a new feature.

STANDARD



Less frequent updating
The suspense of version upgrades



Takes a while to get new features
Upgrades might break stuff

ROLLING RELEASE



At the bleeding edge of software



Updates might break stuff
Constant updating might irritate

STANDARD (POINT)



Debian: The grandfather of Linux distributions. Starting from the year 1992, Debian has been one of the most popular and **stable** distros.

Ubuntu: One of the most popular, beginner friendly Linux distributions (if you ask me, **no**). It is a derivative of Debian and hence uses the same package manager. Has a lot of derivatives of it's own.



Red Hat: The leading enterprise Linux distro. It is where the .rpm package format comes from. Heavily used by businesses.

Fedora: The community version of Red Hat Linux. Free and open source, and uses the .rpm package format. Has several derivatives and even a way to create your own 'remix'.



Elementary OS: A derivative of Ubuntu LTS that is marketed as a lean and privacy focused alternative to Windows and macOS.

Linux Mint: A derivative of Ubuntu that tries to preserve the looks of older desktop paradigms while offering a modern feature set and a complementary suite of apps.



ROLLING RELEASE



Arch Linux: Started in the year 2003 with the goal of being a rolling release distro. Lean and offers a free reign. Is popularly known to be difficult to install and maintain.

Manjaro: A spinoff of Arch Linux that has a graphical installer. One of the easier ways to get into the world of Arch, is now a recommended beginner's distro.



Kali Linux: A rolling release distro based on Debian's testing branch. It's very popular in the world of cybersecurity and pentesting.

openSUSE Tumbleweed: The rolling release version of openSUSE, a distro that uses the .rpm package format. Is the community developed version of SUSE Linux.



Solus OS: A Linux distro that uses it's own package format with the extension .eopkg, and has a minimalist desktop environment.

Credits: myself!

How to enter Open-Source Technologies

- Ankitha Krishnamoorthy , CB.EN.U4CSE18107

Contributing to OSS can be a great way to learn and find like-minded friends. But where does one begin? How to contribute to the world of open source? Who qualifies as a 'contributor'? Well, the happy news is anyone enthusiastic about a certain domain or having even the tiniest bit of experience working with the required skillset can qualify as a contributor! All it requires is just a little bit of research and socializing! (Online communities).

A common misconception around being an OSS contributor is that you need to build a stable software or write 100s of lines of codes to qualify as one. But the truth is that there are so many more dimensions to a contributor: A bug reporter, an innovator who comes up with new ideas for modifications or updates, a maintainer who overlooks a particular fork of the software, a video editor or advertiser who markets the product, a technical writer who helps with product documentation or a polyglot who helps translate these documentations, all qualify as contributors!

Now, here is a funny story of how my friend entered the world of OSS. When we were in our 3rd year, we had been assigned a cool topic for one of our course projects. My friend being a procrastinator didn't bother working on it until a couple of days before her review. And when she did, she simply googled up the project title and lo and behold, an entire GitHub project popped up! The project being public, my friend pulled it, made a couple of changes and presented it and bagged a solid grade. Fast forward a few weeks, I walk into her room to find her completely invested and working on the same project. I then came to know that ever since the review had gotten over, she'd been working on it and discovered how passionate she was about it. She had also managed to find a couple of bugs in the project and reported them!

It's been over a year now but she's still an active contributor to the repository. She's resolved over a dozen bugs and provided efficient algorithms to achieve objectives. But most importantly, she tells me that this past year, she has learnt and grown far more than her peers.

So how can you get into OSS? Well, you have 2 routes: Make your existing projects open-source or, contribute to an open-source project. The first way involves learning the art of structuring your files and codes neatly, making your code readable for any future developer and finally, making the source code open source by hosting it on GitLab/ GitHub and creating a good readme file to explain your work.

The second route revolves around exploring projects that interest you or complement your expertise and tweaking them to ameliorate the project. You could try going to the source code, clicking on the issues tab, learning all sorts of help that the project needs and then getting started! The type of help the project may need may range from documentation to code reviewing, code tidying / optimizing to marketing or simply even just starring the project! The beauty about this is you don't have to be associated with coding or computers in general! All you need is a similar inclination with the vision of the project!

So, that's a wrap, folks! Super excited to see where y'all are going to contribute and what amazing projects are in store for us!



Do's and Don'ts in Open Source

- R Pranav Ajay,CB.EN.U4CSE18140

Open-Source Technologies are awesome in terms of how open, welcoming and inclusive they are, and the fact that they work decentralized makes it even easier for people to get into contributing to a particular project. But we also have to factor in the decorum of contributing to a project. There are always specific guidelines that need to be followed. Now let's explore them briefly as Dos and Don'ts...

Do's:

- Always checkout for the README markdown file in the source repository for tags like "How to Contribute", "Contributing", etc. It explains what the project maintainers and creators are looking for.
- Next head out to the issues section of that project (in GitHub/ GitLab) and look out for all the feature requests and bugs that you could contribute to. Project Maintainers would also label each of the issues with tags.
- Make sure to understand all about the issue/feature request and try to match the deliverables that are mentioned specifically - Expected behaviour, the exact runtime environment and make sure to know the maintainer who gets assigned to monitor the issue.
- Always cherry-pick features from other developers! This isn't kanging and it helps promote inclusiveness and also helps you learn to innovate on existing ideas.

Don'ts:

- Never make a pull request unless you can come up with fixes that fully solves an issue or complete a feature request. Maintainers spend a lot of time reviewing all the pull requests, so you shouldn't waste time and resources by committing half-finished implementations.
- Never expect a maintainer to reply or give a response soon because they aren't contracted to work...but it's out of interest and goodwill for the project, so make sure to be patient and understanding when there is a delay in communication.
- Kanging code isn't an offence as long as the source licenses permit it but make sure to credit the original owner of the portion of code even when they don't explicitly mention that. Not adding authorship in commits where you have a portion of code borrowed from elsewhere is not an acceptable practice.

That's a wrap to the article! I hope you will inculcate all the good things and cultivate them in like-minded folks.

Open Source Projects

UMBRELLA REPORTS

- Siva Kumaran, CB.EN.U4CSE19632

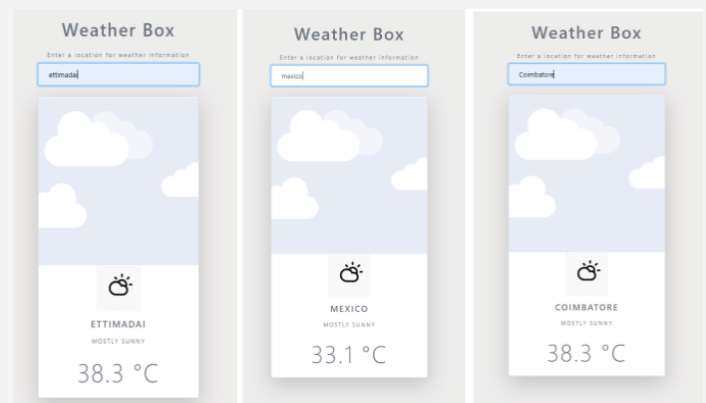


Have you ever felt the need for a versatile site that could tell you the weather conditions of any city along with its climate conditions? Here are the "Umbrella Reports".

This site is capable of telling you the temperature, climate, and weather conditions of any city within seconds. It is a user-friendly website and is currently an open-source GitHub project with 3 contributors. It is one of the repositories in zero to mastery channel.

It also presents a pictorial representation of climate conditions to give a clear view to the users. This is accomplished by using Javascript and API endpoints connected to the actual database. Thus, making the data available in real-time.

REFERENCES: <https://openweathermap.org/api>



POSITION LOCATOR



- Karthik, CB.EN.U4CSE19626

Do you want to locate a place on map using latitudes and longitudes? If yes, then you can use Position Locator. This site uses javascript and leaflet.js website's API to locate the coordinates on map.

Users can enter their desired latitude and longitude and press the submit button in order to get the location of the particular coordinates on a window. Initially, the window is preloaded with the world image in 2d.leaflet.js with the help of the web Mercator method, as we know that world is a sphere and when plotted on 2D it will have some constraints. As leaflet.js uses web Mercator it uses predefined formulas to convert the entered latitude and longitude to coordinates for plotting on image. For zooming purposes tilelayer function is used from openstreetmap website. Then marking points done using marker which is predefined in leaflet library.

REFERENCES: <https://developers.google.com/maps/documentation/api-picker/>

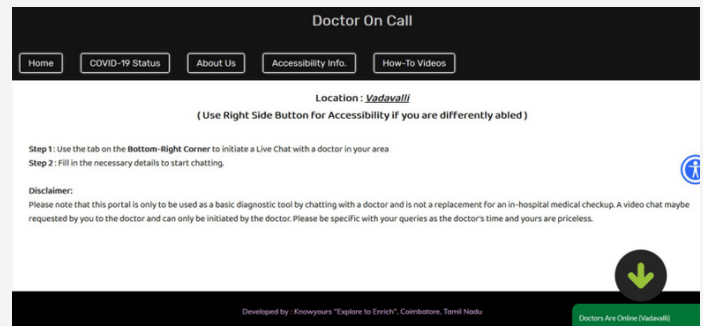
DOCTOR ON CALL

- Sudhay S, CB.EN.U4CSE19652



In this hard time of Coronavirus pandemic. which poses a challenge to the entire human race, people from all fields like national leaders, administrative machineries, medical experts, paramedical & support systems and the police personnel have joined hands to fight against this invisible enemy, COVID-19. It was on the evening of 24th March,2020 when our Prime Minister Narendra Modi announced a nation-wide lockdown for 21 days due to the coronavirus pandemic. The two main things in the announcement that I felt important in this time of national emergency are, the idea of social distancing and the government's request asking people to prevent doing self-diagnosis in lockdown since hospitals don't want patients with mild ailments to make a visit, so as to reduce exposure to the rapidly-spreading virus. These are the core ideas that led towards the development of this free telemedicine platform, Doctor On Call.

This website complies "Telemedicine Guidelines" laid down by Medical Council of India in March 2020. It is highly user-friendly, both for the patient and the doctor. Patients at home can visit the website by clicking on www.doctoroncall.ml



and choose the doctor of their choice from the list of doctors, and interact with them through verbal chat. If the situation warrants a video chat, the doctor can initiate the same through Skype with the due consent of the patient. Once the observation and diagnosis are made by the doctor, the session can be concluded by the doctor writing a prescription online which can be sent to the patient's chat window or by email. There are separate pages designed for the patient and the doctor with respective walkthrough videos paving way to ease of usage of this website. These service has been featured in Dinamalar Daily on 07.04.2020. In just 2 days, 489 people have been benefitted. This service is already put into use by a hospital in Vadavalli. 5 other doctors as on December 2021 are using the platform to attend to their patients by the available modes on the website.

REFERENCES: <https://doctoroncall.ml/>

CHAT ROOM

- Rohit R, CB.EN.U4CSE19644



Have you ever desired a chatting platform , which helps u organize the messages? Here it is the "Chat Room". Just like any other Whatsapp group, this platform helps a user communicate with thier friends. This is accomplished via Firebase, which helps in developing powerful web apps using their complementary web services. Unlike any other

Whatsapp group, this helps a user in organizing the messages. It is very helpful when it comes to searching an old message. Also it helps user to focus on one topic when they are having a communication.

As it is a common platform for all users, for one user to communicate with other they must update their names before sending the message. This is currently a open source GitHub project with 3 contributors and is one of the repositories in zero to mastery channel. Just like any other chatting platform it also operates in real time, i.e., the earlier messages gets stored and succeed the later messages

Three screenshots of a web application titled "Chat Room". The first screenshot shows the initial state with a "Choose a chatroom:" section containing four buttons: "#general", "#gaming", "#music", and "#ninja". Below this are input fields for "Your message:" (containing "Hi") and "Update name:" (containing "Rohit"), each with a corresponding "send" or "update" button. The second screenshot shows the same interface but with a message "Your name was updated to Rohit" displayed below the "Update name:" field. The third screenshot shows the chat history with three messages: "Rohit Hi" (less than a minute ago), "lakshika I have just downloaded Fifa da" (less than a minute ago), and "siva send the file via google drive di" (less than a minute ago). The input fields and buttons are still present at the bottom.

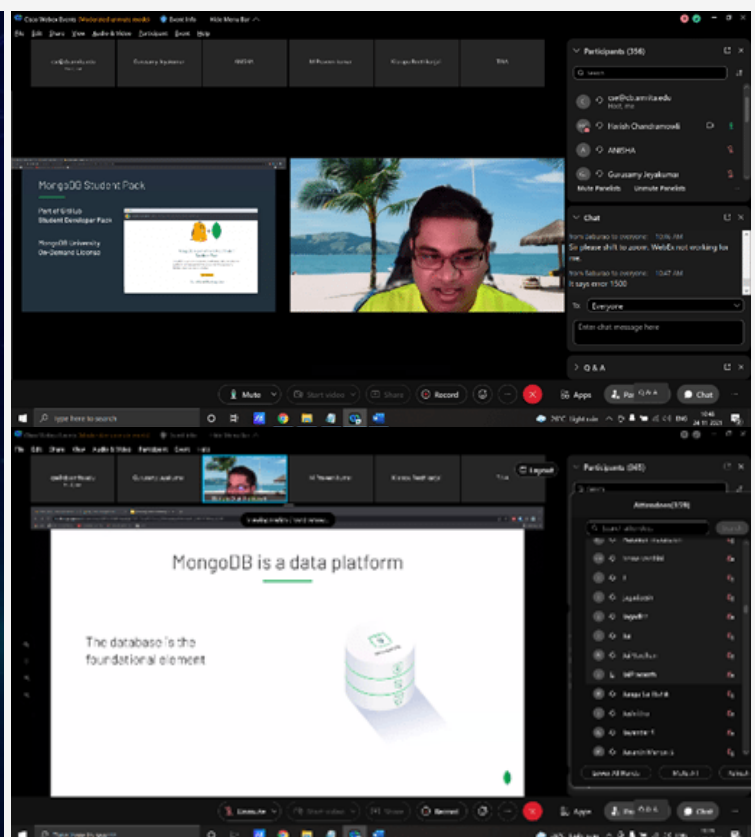
REFERENCES: <https://firebase.google.com/docs>

Workshop on 'Recent trends in databases'

The Department of Computer Science and Engineering, ASE and ASCII conducted a Workshop on Recent Trends in Databases on 24th November, 2021. The event was organized by Bindu K R and Dr. C Arunkumar of the CIR Interaction Team. Reethikanjali, Tina Chowdhary and Praveen were the student coordinators of the workshop. Mr. Harish Chandramouli was the resource person of the event.

The workshop started off with a prayer by Tina Chowdhary. Reethikanjali introduced the guest speaker, Mr. Harish, currently working as a lead engineer at MongoDB, New York, USA. He completed his B.Tech graduation from Amrita University. After dabbling around various concepts, he wanted to explore more in the field of security. He did his masters in the same field at Johns Hopkins University. After finishing masters, he joined Bloomberg as a software engineer to building security features, since his passion was building software over hacking. In his free time, he loves playing board games and working on small fun projects. He is an avid Manchester United fan.

Dr. G Jeyakumar welcomed the guest speaker. The workshop was conducted in Cisco WebEx platform with around 383 enthusiastic B.Tech students attending the same. The event concluded with a vote of thanks.

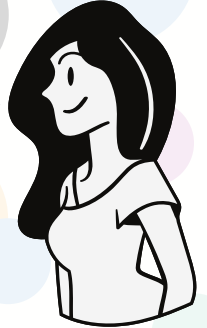


Fun Facts...

- Rohit R, CB.EN.U4CSE19644

A goldfish, at 9 seconds, has a longer attention span than the average internet user. You must deliver the content to the audience in the shortest amount of time possible.

I would rather move on with the internet!



Forgetting to optimize your website for mobile use is an absolute no-brainer. About two thirds of web usage are from smartphones



50% of users tend to open the menu of your site. A nonsensical menu can easily mess with your business.

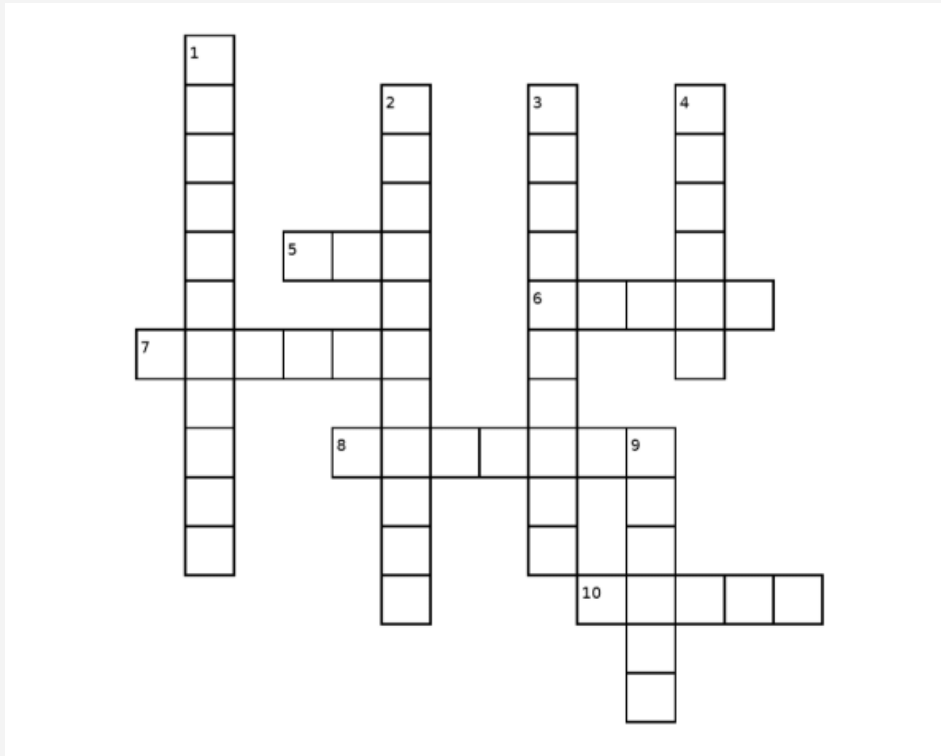


Around 2.9 billion google searches are executed per day. So make sure you get the keywords right!

REFERENCES: <https://www.finoit.com/blog/9-web-application-development-facts/>
<https://www.rd.com/list/brain-teasers/>

Brain Teaser

- Siva Kumaran, CB.EN.U4CSE19632, - Lakshika K, CB.EN.U4CSE19663

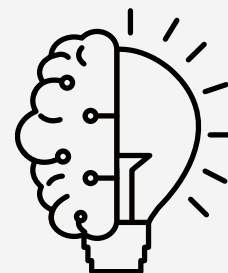
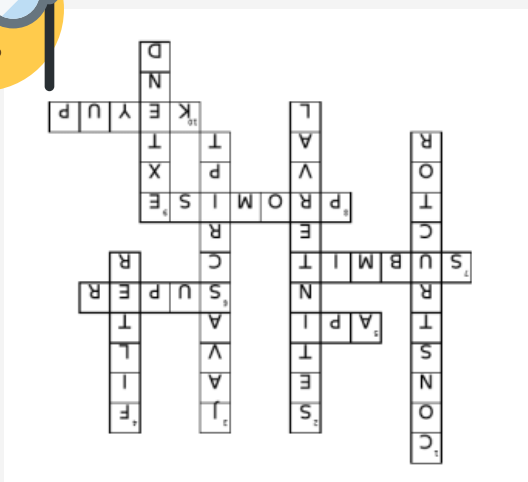


DOWN: ↓

Across: →

- 1) I am present inside all the class
- 2) I am a function, which activates only in certain intervals
- 3) I take care of User Interface
- 4) I am a method in array
- 9) I must be used in inheritance of the classes

- 5) My end point is used as source of the database
- 6) I am used when it comes to looking for parent class constructor
- 7) I am an event listener in form activate at the end
- 8) I am returned by the async function
- 10) I start listening when the key moves up



Student Achievements



In the state Water Polo meet held at Velachery TARUN KB, CB.EN.U4CSE19655 OF computer science department won a gold medal and was the high scorer in the meet. He also represented the Tamil Nadu team as captain. In the Intercampus tournament conducted at Amrita Vishwa

Vidyapeetham. He won the individual champion by securing 5 gold medals in the following events:

- 50 meters breaststroke
- 100 meters breaststroke
- 200 meters breaststroke
- 50 meters butterfly stroke and 200 meters .

He has also took part in the Inter University Waterpolo meet held in Punjab.



Sudhay.S, CB.EN.U4CSE19652 of Computer Science Department is a swimmer who has secured various positions in the district and state-level tournaments in Tamil Nadu. In the inter-campus tournament conducted at Amrita Vishwa Vidyapeetham, he

secured a silver medal in 200 meters Individual Medley, and bronze medals in 100 meters and 200 meters backstroke respectively.

Sudhay is also a world record holder in swimming achieved on 14th November 2016 titled "The Longest Distance Covered using Medley Swimming Technique in 5 Hours" covering a distance of 13.35 Kilometers in 5 Hours duration. The record was adjudicated by 5 World Record companies namely Elite World Records, Unique World Records, Asian Records Academy, India Records Academy, and Tamilan Books of Records.



AMRITA UNIVERSITY INTERCAMPUS CAMPUS

BADMINTON TOURNAMENT WINNER.



Lakshika.K, CB.EN.U4CSE19663

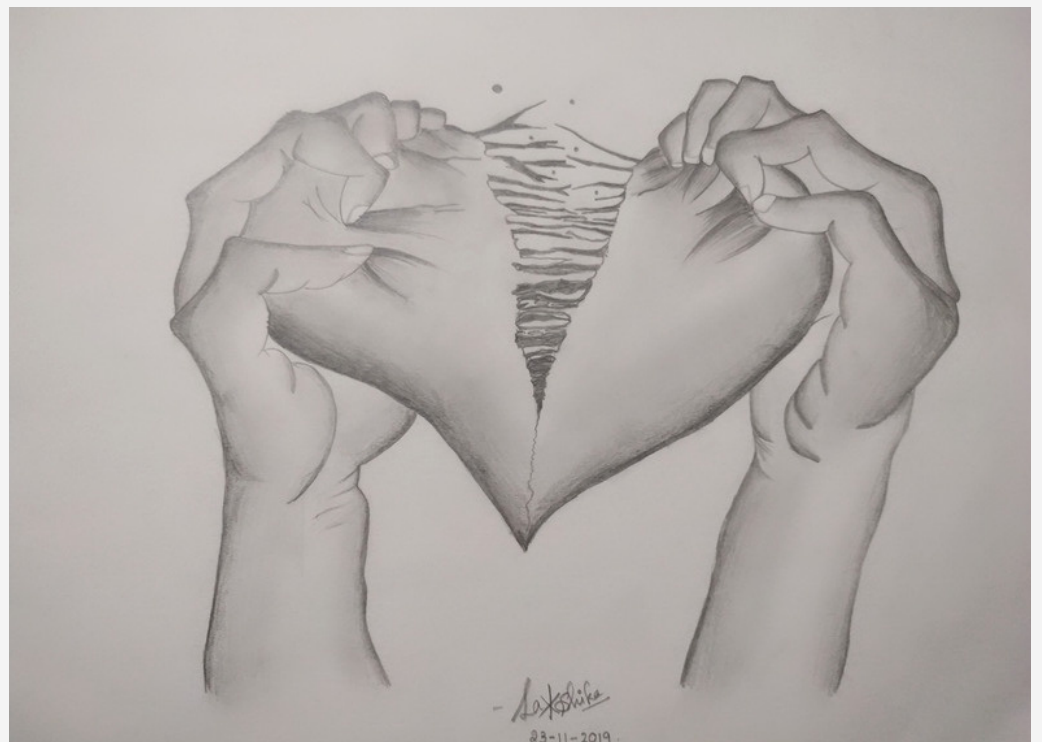
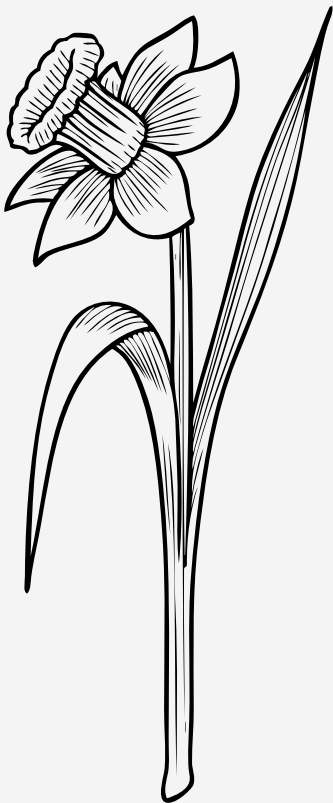


- Represented University in South Zone Inter-University Badminton Tournament to Manipal University, and entered quarter-finals.
- Agneya and Annual Sports Day meet Winner in Badminton singles and doubles
- U-17,U-19 Coimbatore District Champion (Singles and Doubles).
- U-15, U-17, U-19 Zonal level championship (Singles and Doubles).
- Semifinalist in PBL Medlife U-15 at Calicut.
- U-15,U-19 State And Nationals Qualifed.



ART CORNER

Lakshika K, CB.EN.U4CSE19663



EDITORIAL TEAM

FACULTY COORDINATORS

Dr. D.VENKATARAMAN

Mrs. T.BAGYAMMAL

Mrs. JEENA

EXECUTIVE MEMBERS

SAI PRIYADARSHINI, CB.EN.U4CSE17250

ADITHI NARAYANAN, CB.EN.U4CSE18205

ROOPA VIDHYA, CB.EN.U4CSE18143

NIRMAL K, CB.EN.U4CSE19038

MAHIMA LOLLA, CB.EN.U4CSE19128

SUMITHRA S, CB.EN.U4CSE19247

VARADHARAJAN K, CB.EN.U4CSE19257

STUDENT CONTRIBUTORS

ROHIT.R, CB.EN.U4CSE19644



LAKSHIKA.K, CB.EN.U4CSE19663



SIVA KUMARAN.M, CB.EN.U4CSE19632



SUDHAY.S, CB.EN.U4CSE19652



NIRMAL.K, CB.EN.U4CSE19038

