PROGRAMMING CLUB

SUMMER PROJECTS

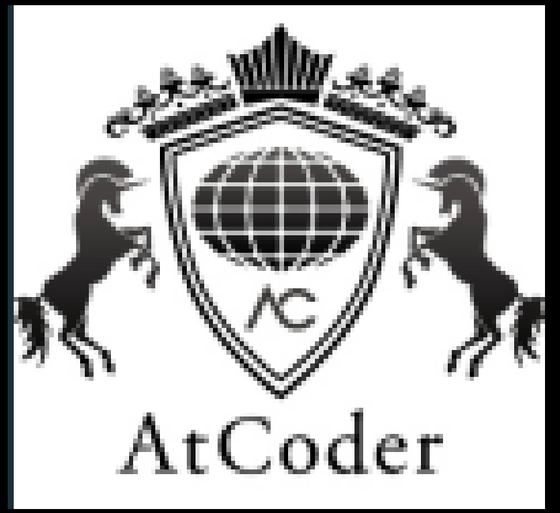
ALGO ARENA

What is Competitive Programming

Competitive Programming is a mind sport where participants need to write programs to solve well-defined problems.

These problems are often related to data structures, algorithms, maths and logic.

Some popular Competitive Programming sites



Why Competitive Programming?

1. It’s fun.

Why Competitive Programming?

2. To understand CP memes and shitposting

What you will be learning

- Greedy Algorithms
- Binary Search
- Dynamic Programming
- A lot of graph related algorithms
- Advanced data structures like Segment Trees and
- Binary Indexed Trees
- The art of Problem Solving!!

LOGISTICS

DIFFICULTY	MEDIUM
DURATION	8 WEEKS
NO. OF MENTEES	30
WORK-LOAD	
MODE	OFFLINE

Prerequisites

- Should be familiar with basic programming concepts like time complexity,binary search and recursion.
- Requires basic C++ knowledge, like familiarity with basic STL containers(vector, pair, set, map, stack, queue).
- Basic knowledge of Probability and Combinatorics.

MENTORS

- DIVYANSH GARG (9818970175)
- JAYANT JHA (8591104712)
- PRERAK AGARWAL (8528203343)
- SWAYAMSIDH PRADHAN (7978339972)

THANK YOU !

MASTERING WEB 3.0

What is Web3.0?

Fed up with Instagram and WhatsApp servers going down? Most internet applications are controlled by centralized entities that determine how they save and use end-user data. Instead of centralized management structures, Web3 (also called Web 3.0, decentralized web, or semantic web) technologies allow for community-driven projects.

ABOUT THIS PROJECT

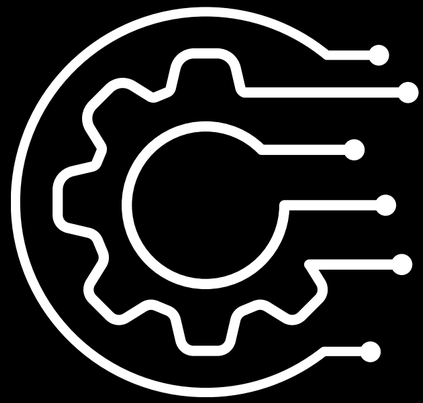
Aiming towards development of blockchain based Web3.0 DApps and implementation of the same. You will learn web3 from scratch and will be able write independent smart contracts and make working DApps.

Final project will be of deploying the IITK-Wordle game, which essentially is a blockchain implementation of the popular game Wordle using but based on campus lingo

WHAT YOU WILL BE LEARNING

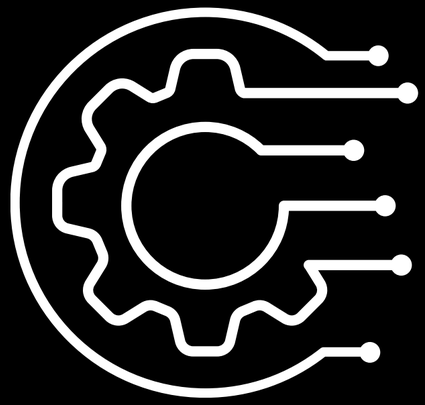
- Blockchain Fundamentals , Ethereum ecosystem, and cryptographic principles
- Hands-on experience in writing smart contracts using Solidity, focusing on essential concepts like data structures, functions, and event handling.
- Frontend development and implementation using HTML , CSS , JavaScript, ReactJS
- Integration of frontend with smart contracts and testing and debugging of the application

LOGISTICS



DIFFICULTY	MEDIUM
DURATION	8 WEEKS
NO. OF MENTEES	20
WORK-LOAD	15-20 (hours per week)
MODE	OFFLINE

PRE-REQUISITES



- Should be familiar with Object Oriented Programming(OOPs) in C++
- Basic Knowledge of frontend development

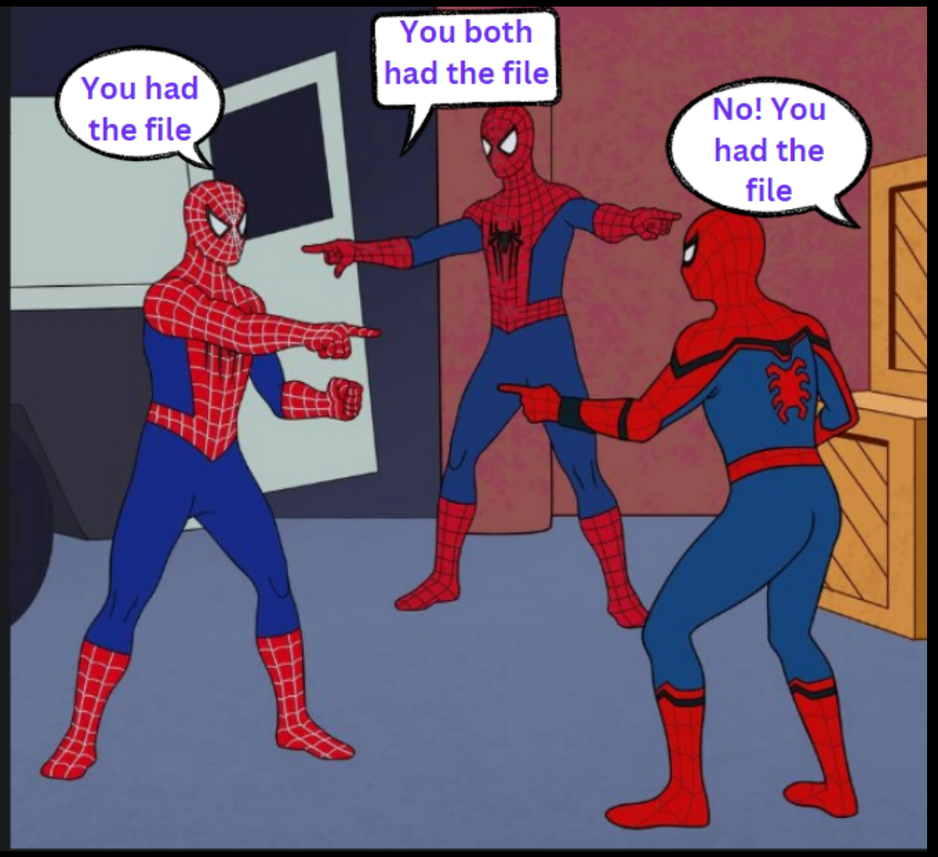
MENTORS

- MAYANK AGARWAL(9289077004)
- TAMOGHNA KUMAR(7619186444)
- CHAYAN KUMAWAT(9569426190)
- KINCHIT GOYAL(8619130018)

THANK YOU !

NFS GOES ONLINE

WHAT WILL HAPPEN IN THE PROJECT?



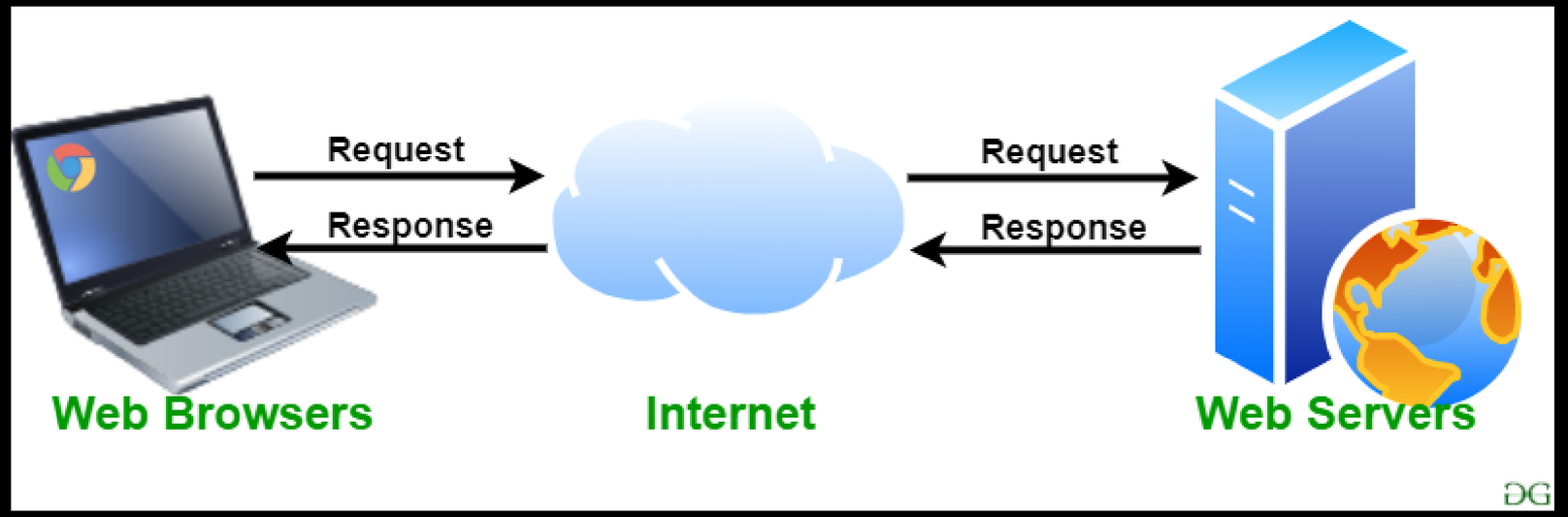
In this project, you will get introduced to low-level programming and dive into fundamentals of systems and networking.

WHY THIS PROJECT?

- Will help gain a comprehensive understanding of the underlying infrastructure of computer networks.
- provide hands-on experience in applying theoretical concepts learned from textbook
- Proficiency in systems and networking is in high demand in the job market. (So, might help you grab an high-paid intern)

WHAT WILL WE DEVELOP IN THE PROJECT?

A Web Server Written in Rust



A Distributed Filesystem Written in C

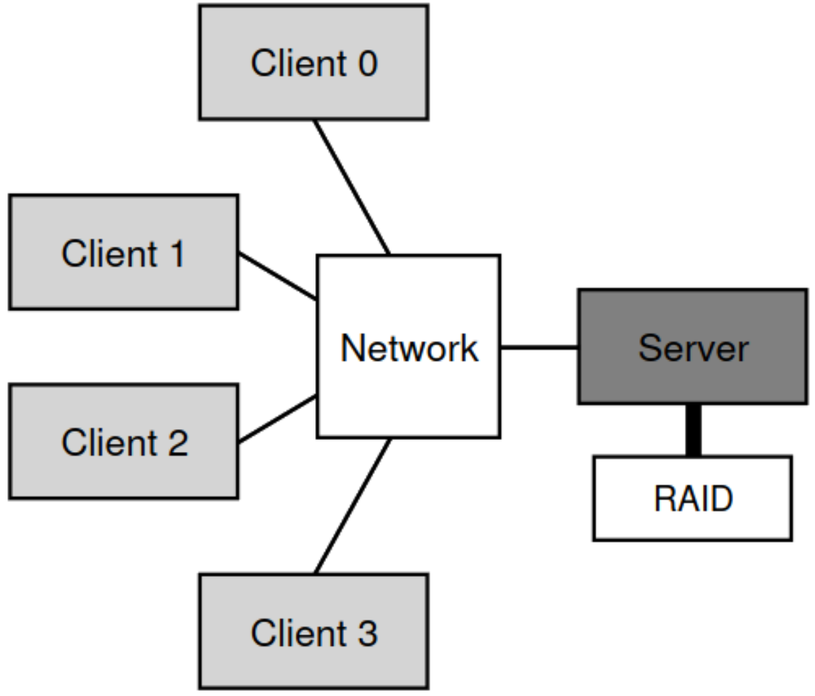


Figure 49.1: **A Generic Client/Server System**

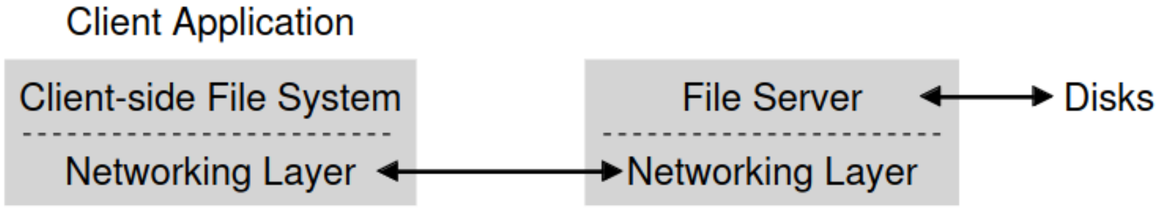


Figure 49.2: **Distributed File System Architecture**

More about the network file system

1. Will be following the well-known OSTEP Book (Operating Systems - Three Easy Pieces by Arpaci-Dusseau).
2. File system disk will be that of a simple flat file, which will be accessed using various system calls. Will be implementing the Unix File System (UFS) which is a very simple filesystem, but still contains all the functionality required by a filesystem.
3. Network wrappers will be written using UDP. The functionalities available to the end user will be stateless and similar to that of Sun Microsystems' Network File System (NFS), which was the first commericially successful distributed filesystem.

More about the network file system

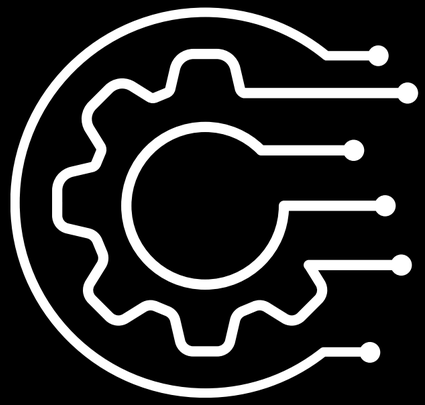
1. Client-side library will be shipped as a static library linked by developers who want to use the system. As such, mentees will also be taught basics about the compilation process, makefiles, etc.
2. You will learn the various difficulties in systems programming and system design - error handling and memory leaks for eg, and how tools such as valgrind and gdb can be a savior.
3. Will deploy and run the system on various computers at the end.

LOGISTICS

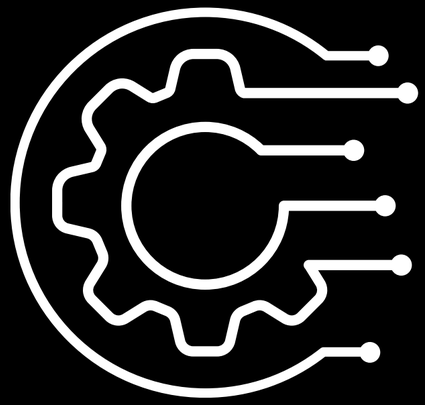
- **DIFFICULTY** — TOUGH
- **DURATION** — 8 WEEKS
- **NO. OF MENTEES** — 15
- **WORK-LOAD** — 

(hours per week)
- **MODE** — OFFLINE

PRE-REQUISITES



1. Basic Rust (upto chapter 3 of Rust Book)
2. Linux files basics (know about the file structure, such as what are files and directories, basic commands related to the same such as cd, rm, mkdir, ls etc.)
3. C language (upto ESC112)



MENTORS

- HARDIK JINDAL (9582890169)
- ASHISH AHUJA (9663712367)
- RITVIK GOYAL (9981114769)
- SHIVANSH MANGAL (9780581811)

THANK YOU !

IITK MAIL CLIENT

ABOUT THE PROJECT

We'll be aiming to develop a mail client for IITK which would provide the campus junta with a reliable and efficient means of communication within the institute community.

WHY DO WE NEED A NEW MAIL CLIENT?

- With the closure of MyMail, there's a pressing need for a suitable alternative at IIT Kanpur.
- While alternatives like Edison-software email exist, a custom mail client offers tailored functionality superior to web-based platforms like webmail.
- Customization is crucial; we can adapt our mail client to the specific preferences of the campus community for optimal efficiency.
- Developing our own client enables brainstorming and implementation of unique features catered to the distinct requirements of IITK.
- Plus it would be cool...right?

WHAT YOU'LL BE LEARNING

01. You'll be learning mobile app development using flutter and dart for both android and ios

02. You'll get to know about how mail clients work, email protocols like smtp, imap/jmap, pop3 and many more

03. You'll be gaining hands on experience in building a professional product which would be used by many users across the campus

04. You'll be learning real life product testing and app deployment and would be credited for the development :)

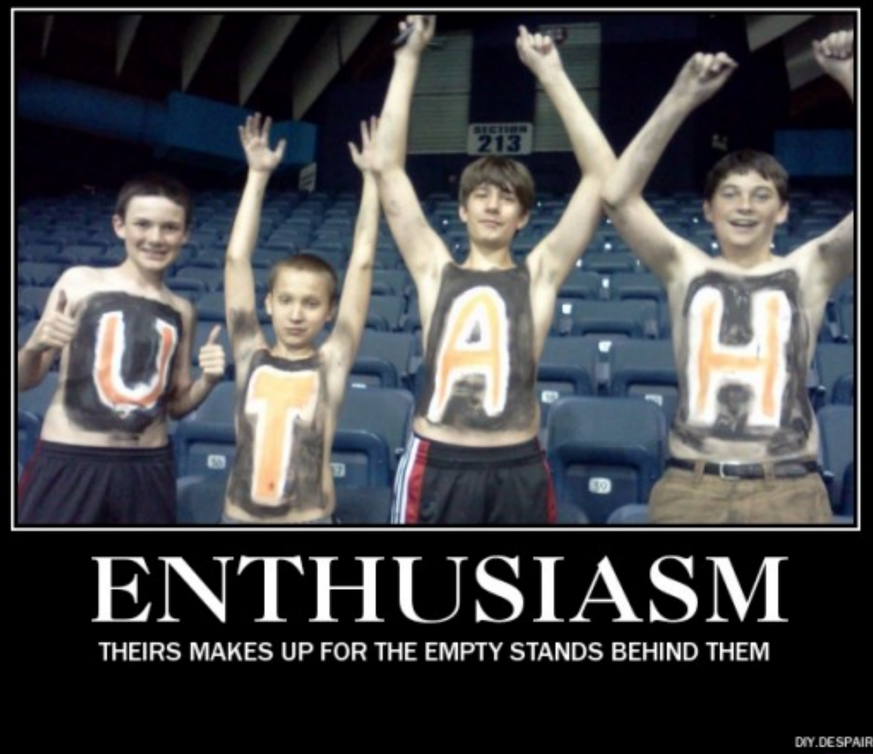
SCREENING FOR THE PROJECT

SELECTION TASK

Tasks for frontend/backend using any stack, revolving around mail client

PRE-REQS

Basic development knowledge would suffice. Development using flutter would be a plus (not necessary).



LOGISTICS

DIFFICULTY	TOUGH
DURATION	8 WEEKS
NO. OF MENTEES	12
WORK-LOAD	 (hours per week)
MODE	OFFLINE

MENTORS

- BURHANUDDIN MERCHANT (8815992866)
- YASH CHAUHAN (9837090171)

THANK YOU !

VERSION CONTROL SYSTEM


ONE DOES NOT SIMPLY
UNDERSTAND GIT

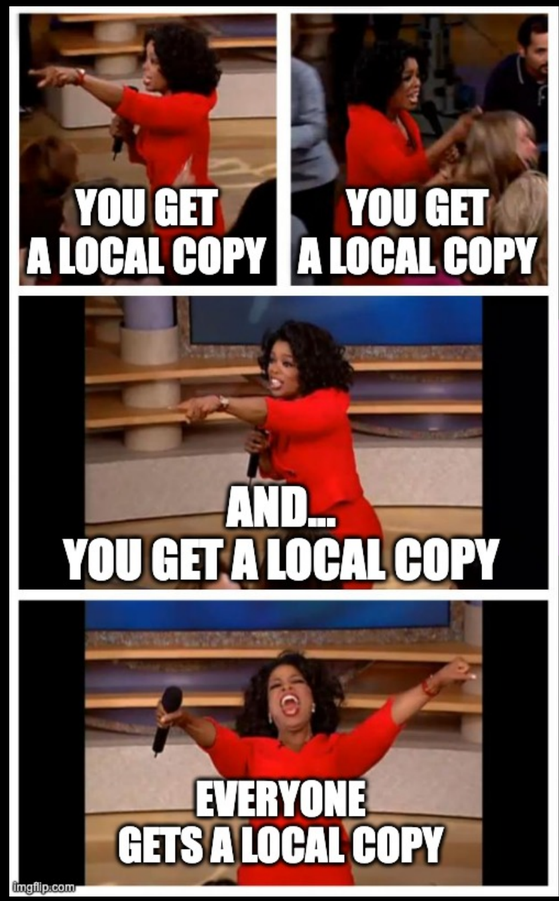
In this Project, You will be learning

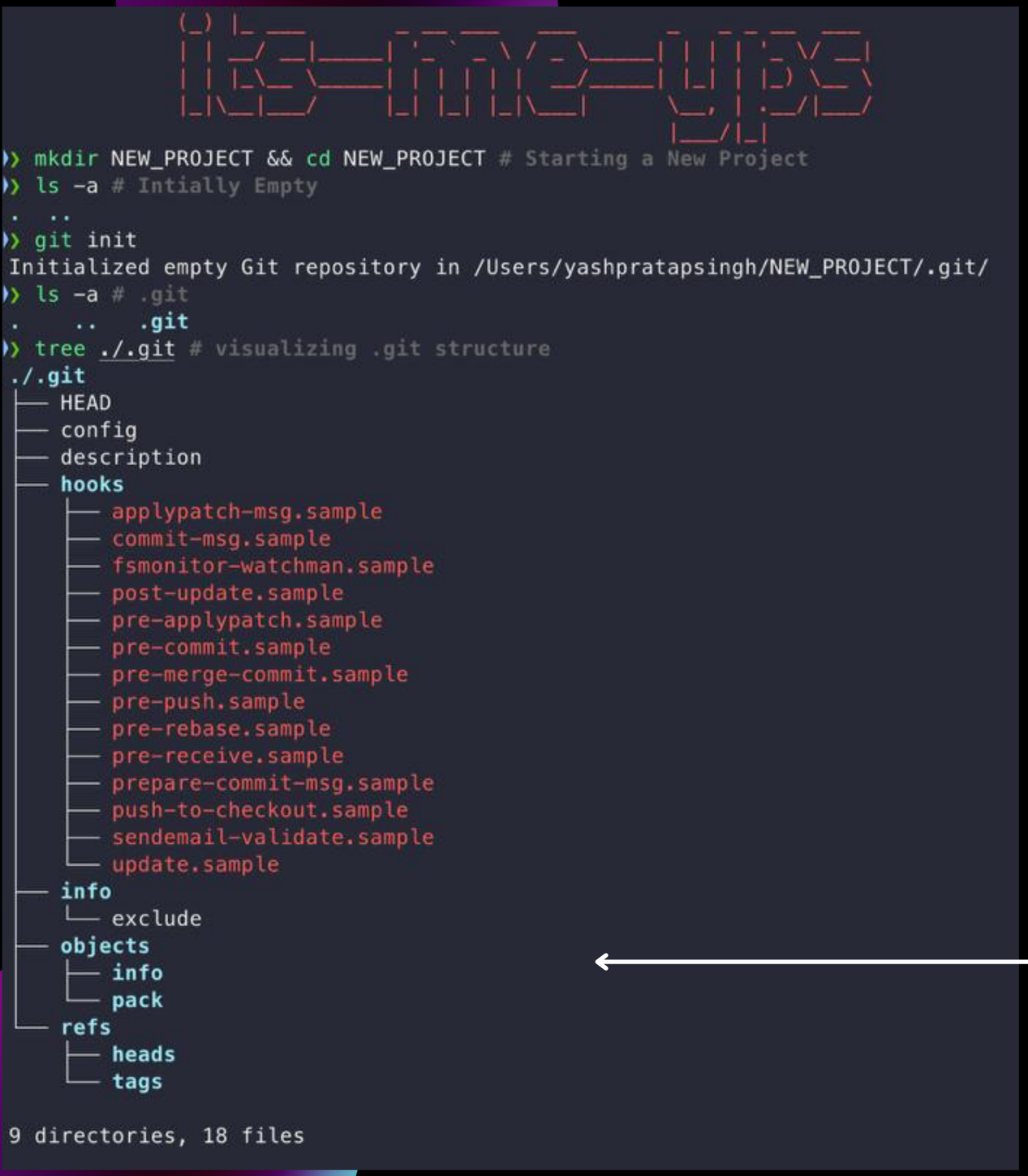
- What is a Version Control System ?
- How to properly use Git ?
- How to make one on your own ?
- and some cryptography and file compressing also.

What will we be building ?

Our own Git like Version Control System, which will be compatible with Git.

And a Distributed Version Control system like Github





```
❯ mkdir NEW_PROJECT && cd NEW_PROJECT # Starting a New Project
❯ ls -a # Intially Empty
.  ..
❯ git init
Initialized empty Git repository in /Users/yashpratapsingh/NEW_PROJECT/.git/
❯ ls -a # .git
.    ..   .git
❯ tree ./.git # visualizing .git structure
./.git
├── HEAD
├── config
├── description
├── hooks
│   ├── applypatch-msg.sample
│   ├── commit-msg.sample
│   ├── fsmonitor-watchman.sample
│   ├── post-update.sample
│   ├── pre-applypatch.sample
│   ├── pre-commit.sample
│   ├── pre-merge-commit.sample
│   ├── pre-push.sample
│   ├── pre-rebase.sample
│   ├── pre-receive.sample
│   ├── prepare-commit-msg.sample
│   ├── push-to-checkout.sample
│   ├── sendemail-validate.sample
│   └── update.sample
├── info
│   └── exclude
├── objects
│   ├── info
│   └── pack
└── refs
    ├── heads
    └── tags

9 directories, 18 files
```

```
❯ echo "version control system" > project.txt # Creating a new txt fi
❯ echo "Aman Sameer Shreya Yash" > mentor.txt # Creating a new txt fi
❯ git add .
❯ git commit -m "First Commit"
[main (root-commit) 4dbf089] First Commit
 2 files changed, 2 insertions(+)
 create mode 100644 mentor.txt
 create mode 100644 project.txt
```

2 git add and git commit

1 Contents of .git after running git init

```
❯ tree ./.git # visualizing .git structure
./.git
├── COMMIT_EDITMSG
├── HEAD
├── config
├── description
├── hooks
│   ├── applypatch-msg.sample
│   ├── commit-msg.sample
│   ├── fsmonitor-watchman.sample
│   ├── post-update.sample
│   ├── pre-applypatch.sample
│   ├── pre-commit.sample
│   ├── pre-merge-commit.sample
│   ├── pre-push.sample
│   ├── pre-rebase.sample
│   ├── pre-receive.sample
│   ├── prepare-commit-msg.sample
│   ├── push-to-checkout.sample
│   ├── sendemail-validate.sample
│   └── update.sample
├── index
├── info
│   └── exclude
├── logs
│   ├── HEAD
│   └── refs
│       └── heads
│           └── main
├── objects
│   ├── 4d
│   │   └── bf08933bebc5e8d5802a840d81b7f3ba8b646e
│   ├── a7
│   │   └── be9ea2beb7ae23229f2f3ad41b613a1f126a37
│   ├── c5
│   │   └── 0a9fc69de838e45606f214b50b6c8b2dd9dd6f
│   ├── e1
│   │   └── daefda1d01632b6e6b9c619f5b8648e0ce3b4e
│   ├── info
│   └── pack
└── refs
    ├── heads
    │   └── main
    └── tags

❯16 directories, 27 files
```

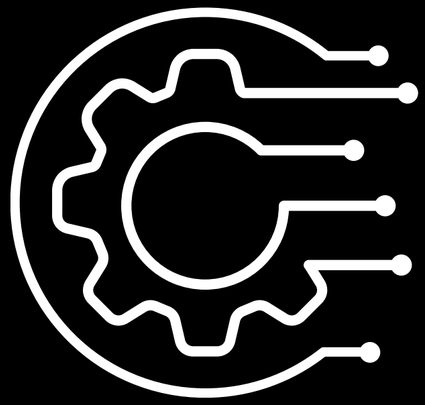
Analysing and understanding how git works and then trying to implement it from scratch.

- Merge Algorithm
- Conflict Resolution
- Data Compression

3 Contents of .git after commit

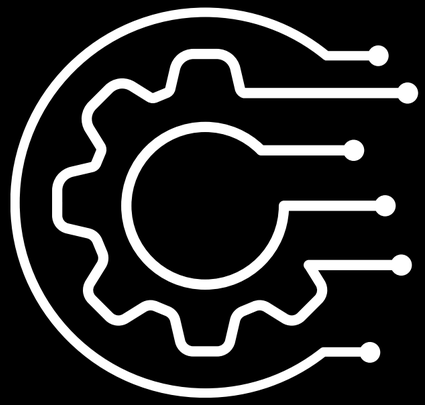
LOGISTICS

DIFFICULTY	TOUGH
DURATION	7 WEEKS
NO. OF MENTEES	10
WORK-LOAD	 (hours per week)
MODE	OFFLINE



PRE-REQUISITES

1. Basic Data Structures
2. Basics of Golang
   (Go by Examples)
3. Basic shell commands
   (CLI for begineers)



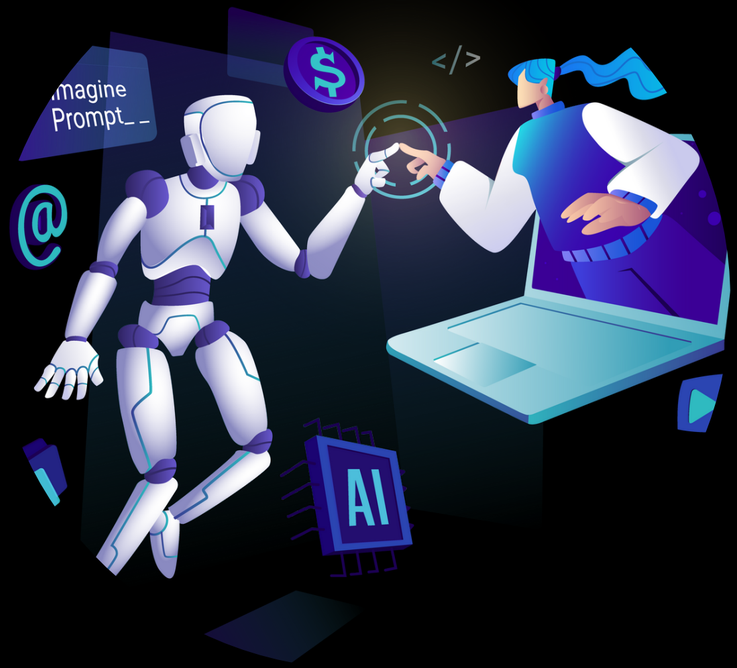
MENTORS

- AMAN SINGH GILL (9911262261)
- SAMEER YADAV (9462392749)
- SHREYA SHREE (7301211451)
- YASH PRATAP SINGH (7007519537)

THANK YOU !

ILLUSION CRAFT: A DIVE INTO ARTIFICIAL CREATIVITY

UNDERSTANDING GANs (GENERATIVE ADVERSERIAL NETWORKS) AND ITS VARIATIONS AND EXPLORING THEIR POWER



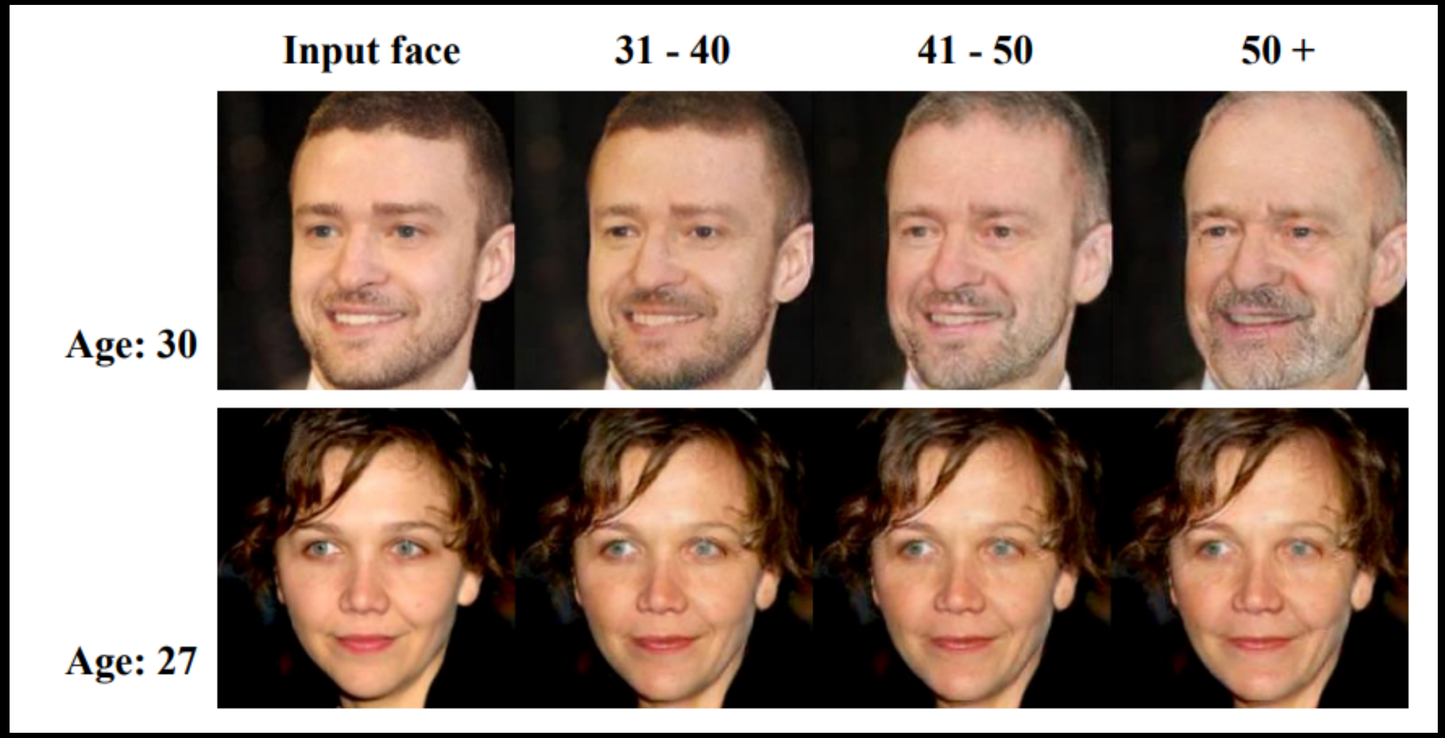
By the end of the project, we aim to develop :-

A TEXT TO IMAGE GENERATION MODEL CAPABLE OF GENERATING IMAGES BASED ON TEXT PROMPT

A small cactus wearing a straw hat and neon sunglasses in the Sahara desert.

An alien octopus floats through a portal reading a newspaper.

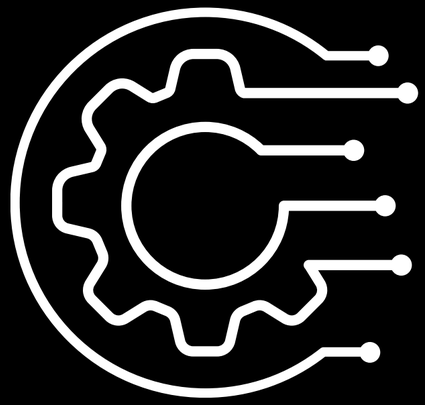
AN AGE TRANSFORMATION MODEL CAPABLE OF AGING AND DEAGING IMAGES



LOGISTICS

DIFFICULTY	TOUGH
DURATION	8 WEEKS
NO. OF MENTEES	12
WORK-LOAD	 (hours per week)
MODE	OFFLINE

PRE-REQUISITES



1. Basic Linear Algebra
2. Basic Python Programming
3. Basic Machine Learning
(Linear Regression and Gradient Descent)
4. OOPs Concepts (Recommended but not compulsory)

MENTORS

- ABHISHEK KUMAR (8240633513)
- RUDRADEEP DATTA (7044845595)
- RAGHAV MANGLIK (8595142080)
- ABHIJIT SINGH JOWHARI (7354190828)

THANK YOU !

Capture The Flag

WHAT ARE CTF'S ?

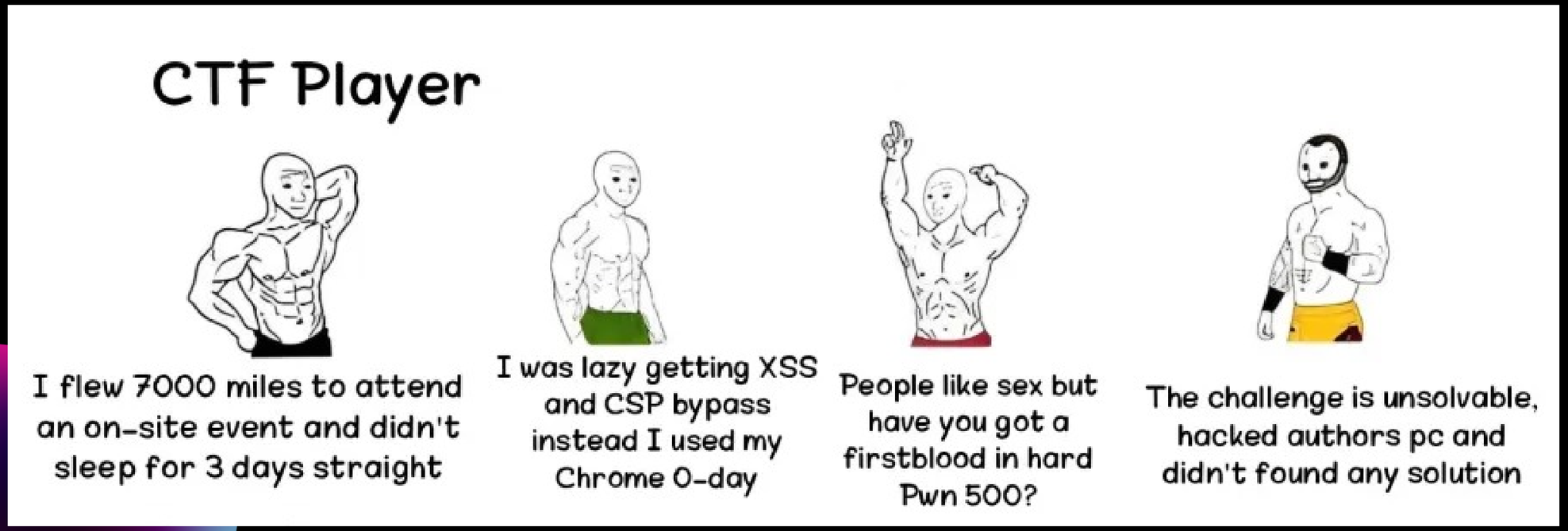
CTFS, OR CAPTURE THE FLAG EVENTS, ARE GAMIFIED COMPETITIONS IN CYBERSECURITY. INDIVIDUALS OR TEAMS COMPETE TO FIND HIDDEN PIECES OF DATA, CALLED FLAGS, WITHIN A SIMULATED DIGITAL ENVIRONMENT.

CTFS ARE SUPER FUN

- Some challenges require that you learn something new that you have never met before.
- You will develop a hacker mindset (finding vulnerabilities in your own programs).
- There are cash prizes, swags, Conference tickets , and most importantly the learning

BUT WHAT THIS PROJECT ?

PARTICIPATING IN MULTIPLE CTF'S & GAINING LEARNING OF MULTIPLE DOMAINS (PWN,REV,CRYPTO,WEB,STEGO)



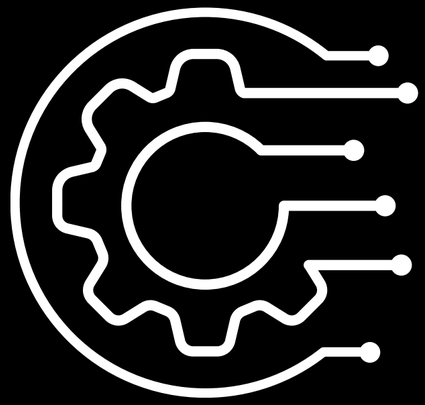
WHAT WILL HAPPEN IN THE PROJECT?

- Participating in CTFs and maintaining all write-ups on a website.
- Team based Evaluation (Competitive environment oh ya)
- Doubt & Lecture Sessions , will be discussing the weak topics and the unsolved flags.

LOGISTICS

DIFFICULTY	Tough
DURATION	8 WEEKS
NO. OF MENTEES	15
WORK-LOAD	 (hours per week)
MODE	Flexible (Online & Offline)

PRE-REQUISITES



1. Linux fundamentals.
2. C language (upto ESC112).
3. Basic Web Dev.
4. Pclub Infosec Roadmap.
5. Basic Python language (optional)

MENTORS

- RITVIK GOYAL (9981114679)
- AMAN SINGH GILL (99112 62261)

THANK YOU !