Machine Learning and Deep Learning:

1. **Audio Description for visually impaired person (Filled - Mentor cannot take anymore teams for this project)**

   Mentor: Harsh Walia
   Tech stack: Python
   Difficulty: Medium/Hard

   Description:
   This project is about converting the image ->caption -> Audio.
   In this project, we have to input the image into model that will generate a one line caption of the image and then generate the audio from that caption.

   Prerequisite knowledge:
   Python, understanding of Machine Learning.

2. **Optimising Cloud Computing Resources Allocation: Leveraging Load Balancer equipped with LSTM Deep Network**

   Mentor: Aadit Agarwal
   Domain: Cloud Computing, Deep Learning, Distributed Systems
   Tech stack: Python, WebShark, TensorFlow, AWS / GCloud / Azure (Choice of mentee)
   Difficulty: Medium/Hard

   Description:
   The improvement of cloud computing by optimizing resource allocation and other related optimizations has been extensively researched and worked on over the past few years. Load balancing, one of the major difficulties that requires special attention, is one of many problems that cloud computing research is now confronting. To identify the optimum solution for enhancing cloud resource consumption, a number of other challenges, including optimizing resource allocation with Machine Learning techniques and minimizing ML load, virtual machine (VM) migration, VM security, user QoS satisfaction, and resource use, need to be given equal consideration.
   Features:

- ➔ Optimize and efficiently handle the payload of server requests along side the processing of the Machine Learning optimizations using a Load Balancer with a Concurrent Hashing algorithm.
- ➔ The proposed solution also leverages the advantages of Machine Learning optimizations by implementing a Recurrent Feedback based Deep Neural Network of Long Short-Term Memory Deep Networks.

Prerequisite knowledge:
Deep Learning, Python, Tensorflow, Knowledge of Distributed Systems / Cloud Computing would be beneficial

3. **Spotify Hit Predictor (Filled - Mentor cannot take anymore teams for this project)**

   Mentors: Ashok Arora
   Tech stack: python, scikit-learn, numpy, pandas, plotting library such as matplotlib, HTML, CSS, React
   Difficulty: Medium/Hard

   Description:
   Mentor has divided the project into two parts, a frontend and a ML part. Depending on their familiarity with the tech stack, mentor will guide them with the particular part. This project's purpose is to create a binary classification model that can predict whether a Spotify song will be a hit or a flop. There will be a web interface that allows users to search for songs, and then we will extract the features of those songs and use them for classification using the Spotify API. The model should correctly predict the class, and we will train our classification model using the Billboard Hot 100 as a benchmark.

   Prerequisite knowledge:
   Python, HTML and CSS. It's good if they have ML knowledge otherwise I can teach them during the project too.

Web Development:

4. **CineStocks**

   Mentors: Palak Jain, Parth Lawania

Tech stack: MERN, Capacitor JS, Ionic
Difficulty: Easy

Description:
The project will consist of an admin panel and a cross platform app.

Producers can tokenize their movies, and then those token will be made available for users for trade. Users can get returns on their investment, once the movie releases in the theaters.
Any other cross-platform app ideas are welcome.

Prerequisite knowledge:
One should have at least basic knowledge of the MERN stack.

5. **REST API Development Library**

Mentors: Aaryak Shah, Harsh Sharma, Asutosh Paramar
Tech stack: Web Development, Library/Framework Development
Difficulty: Medium/Hard

Description:
Design and develop a library or framework that can be used to write REST APIs, similar to Express.js, Flask, or even something of a higher scope like Django.
Features:

- ➔ Routing
- ➔ Middleware
- ➔ Cookies, Headers and Status Codes
- ➔ Handling Response/Request Types (String, Bytes, JSON, FormData, etc.)
- ➔ Feel free to make your own additions to these core features of the library.

Prerequisite knowledge:
You can write a library for any language of your choice. Common preferences are JavaScript, Python, or Java. But we encourage you to try out something different like Dart or Rust... or even C++ if you're up for the challenge.

6. **TrustVault: A Secure Data Storage and Sharing Platform**

Mentors: Anirudh Gautum,Aaryak Shah
Tech stack: Web Development
Difficulty: Medium/Hard

Description:
TrustVault is a web-based platform that allows users to store and share sensitive data, such as financial information and personal messages, with nominated individuals. The platform continuously monitors the status of the user and, upon reaching a predetermined threshold, automatically delivers the stored data to the nominated individuals.
Features:

- ➔ Routing
- ➔ To provide users with a secure and convenient way to store and share sensitive data with trusted individuals.
- ➔ To give users peace of mind by automating the process of sharing important information in the event of an emergency or unforeseen circumstance.
- ➔ Users will sign up for a TrustVault account and create a profile.
- ➔ They will then be able to upload and store data on the platform, including financial information, personal messages, and other important documents.
- ➔ Users can also attach nominated individuals to their stored data, specifying who should receive the information in the event of an emergency.
- ➔ The platform will continuously monitor the status of the user and, upon reaching the predetermined threshold, automatically deliver the stored data to the nominated individuals.

Prerequisite knowledge:
This is a language and framework-independent project. Students are free to choose the language or framework. Priority will be given to implementing best practices for API security and data storage throughout the development process.

7. **Cryptographically Secured Password Manager (Filled - Mentor cannot take anymore teams for this project)**

Mentors: Kunal Jain
Tech stack: Web Development, Cryptography
Difficulty: Medium/Hard

Description:
We'll be building a password manager that'll securely store user-entered passwords and usernames/emails by cryptographically encrypting them. This project will require some base knowledge of JavaScript/TypeScript and a bit of intro to any development in general.
Features:

- ➔ Building a responsive user interface to make the features available for the end user.
- ➔ Using public and private key based cryptography to store user credentials securely.
- ➔ Modularity and scalability of code.
- ➔ Following proper version control flow to maintain the project and collaborate.

Prerequisite knowledge:
The project has some prerequisites because that'll help build the entire platform at a proper pace, but if you're someone willing to learn things from scratch and give it a go, do join and would love to help!

8. **CLI Based Key-Value Store (NPM Package) (Filled - Mentor cannot take anymore teams for this project)**

Mentors: Kunal Jain
Tech stack: Packages, CLI, Backend Development
Difficulty: Medium/Hard

Description:
We'll be building a command line interface that'll be published on NPM. Using the command line tool, you can authenticate yourself as an existing user or new user and store data in the form of a key-value store directly from your terminal across different operating systems and devices. The project requires some basic knowledge of token-based authentication systems and some experience working with databases and writing a backend server.
Features:
- ➔ CLI-based authentication using Auth0 or some other alternative.
- ➔ Building a backend server that remains in sync with Auth0 to respond to authenticated user requests and store their key-value pairs in the database.
- ➔ Building an interactive CLI for the user to access the entire application.
- ➔ Making the CLI accessible to multiple operating systems.
- ➔ Modularity and scalability of code.
- ➔ Following proper version control flow to maintain the project and collaborate.
- ➔ Using public and private key based cryptography to store user credentials securely.

Prerequisite knowledge:
The project has some prerequisites because that'll help build the entire platform at a proper pace, but if you're someone willing to learn things from scratch and give it a go, do join and would love to help!

9. **<u>E-commerce Web Application</u> (Filled - Mentor cannot take anymore teams for this project)**

   Mentors: Varun Kumar Tiwari, Shikhar Gupta
   Tech stack: Web Development
   Difficulty: Medium/Hard

   Description:
   Build an e-commerce web application to sell various categories of electronics such as laptops, desktops, storage devices, audio devices, etc.
   Features:
   - ➔ User authentication: Use email-password authentication as well as use oAuth providers such as Google, Facebook, Twitter, etc.
   - ➔ Build a front end using either Next.js or React.js.
   - ➔ Build a backend using Node.js.
   - ➔ Integrate checkout using either Stripe, Razorpay, or Paypal.
   - ➔ Build an admin panel to manage the application.

   Optional Features:
   - ➔ Multi-vendor support.
   - ➔ Email notifications.
   - ➔ Simple recommendation system using past orders and cart data.

   Prerequisite knowledge:
   Basic Web Development Concepts

10. **<u>Movie Voting Platform</u> (Filled - Mentor cannot take anymore teams for this project)**

    Mentors: Shambhavi Shandilya, Soumya Singh
    Tech stack: Preferably MERN stack. With any alternatives to the database and any additional services like Firebase or Lambda functions (optional).
    Difficulty: Medium/Hard

    Description:

A full-stack project where admins can create award contests and users can vote for their favorite movies.

Features:

- ➔ Displaying movies/ tv series and their details (Free APIs such as TMDB or IMDB can be used).
- ➔ Adding the functionality to search/filter movies.
- ➔ Handling granular logging system: There will be three types of users. Owners: Users who can create contests/ blacklist movies from a contest. Artists: Users who can add their own movies (apart from the API-provided movies). Audience: Users who will vote in contests.
- ➔ The contests can be manually started and ended by the owners. The audience can vote for up to 5 movies each in a contest.

Prerequisite knowledge:
Basic knowledge of web development. You can create a simplified version of this project to get started with web development.

11. Homecast

Mentors: Varun Kumar Tiwari, Shikhar Gupta
Tech stack: Web Development
Difficulty: Medium/Hard

Description:
HomeCast is a media server application that allows you to organize and stream your media files, such as music, movies, TV shows, and photos, from a central location to any device with a browser client installed. Users can create a centralized library of their media and access it from any device that is connected to the same network. The server can be accessed through a web browser that is available for a variety of devices, including smartphones, tablets, and smart TVs.

Features:
1. Create/manage media libraries.
2. Stream media from the local system to the home network.
3. Automatic organization and creation of metadata.
4. Persist watch history.

Optional Features:
1. Custom video player/controls.
2. Simple recommendation system.

3. Multiple accounts: The hosting account has all the privileges. Other accounts need to take permission from the hosting account to view or stream any content.

Example application: Plex Media Server
(https://www.plex.tv/media-server-downloads/)

Prerequisite knowledge:
Basic Web Development Concepts

12. **BookMySpot (Filled - Mentor cannot take anymore teams for this project)**

Mentors: Manish Kumar
Tech stack: Frontend - HTML/CSS/ReactJS, Backend - Node.JS / Express, Languages - JavaScript/Typescript
Difficulty: Medium/Hard

Description:
Build an interactive platform to book time period based spots at certain points on the map of a region or area, anonymously. It's a fun project, probably won't have that much impact, but there will be a huge amount of learning for sure.

Prerequisite knowledge:
Primary prerequisite is their willingness to learn, grind, and commit. Some web development knowledge would be good, but make sure they know that it's not a requirement or necessity

13. **Remote Code Executor / Zoom transcriber / Open to idea**

Mentor: Hiten Sharma, Kavish Dadhich
Tech stack: MERN Stack
Difficulty: Medium/Hard

Description:
Building a code execution platform like CodeChef IDE along with some extra features like safe code execution.

Prerequisite knowledge:
Building web apps with MERN stack. Some Docker knowledge would be nice to have for extra features.

14. **University Broadcasting Application**

Mentor: Sankat Mochan
Tech stack: react + next for frontend, node + express for APIs and go+ gorilla+ gin for websockets
Difficulty: Medium/Hard

Description:
This application will provide a platform for the universities to send and receive messages securely.
Features:

- ➔ Basic CRUD operations on messages, websockets using socket.io for realtime conversation
- ➔ building own websocket server using go and gorilla(if time permits)
- ➔ one on one chat
- ➔ group chat
- ➔ allowing users to send media files in chat
- ➔ normal authentication using otp and oauth.

Prerequisite knowledge:
Some previous experience in MERN stack is required

15. **Food Delivery Consumer Mobile Application**

Mentors: Swarnim Gupta, Harshit Yadav
Tech stack: React Native, Node.js, Firebase
Difficulty: Medium/Hard

Description:
We'll be making a food delivery mobile app for a consumer ordering the food item and will try to add all the features that it must have during the course of the winter projects.
Features:

- ➔ Authentication
- ➔ Cart
- ➔ Payment Gateway
- ➔ Global Search
- ➔ Map Integration
- ➔ Realtime tracking system (Optional - Separate Microservice)
- ➔ Local and Broadcast Notification

Prerequisite knowledge:
React(Beginner), Javascript

16. Data Collection Platform

Mentors: Thari Zephaniah
Tech stack: Web Development
Difficulty: Medium/Hard

Description:
The data collection lifecycle via any good platform does not end with the submission of a response. There is usually some post-submission business logic that needs to be supported over time, like sending an email to the user once they submit the form, sending an SMS, or adding data into a google spreadsheet. Consider it an advanced version of google forms.
Our project will focus on building a web application with significant aspects/actions like sending Sms/sending or email and deploying it to the web.
Features:

- ➔ Develop a form builder which allows the users to add questions with different option types, along with the ability to submit a response to the form.
- ➔ Tracking form submissions
- ➔ Sending a response mail/sms to the user who submitted the form
- ➔ Export form data in a CSV file.

Prerequisite knowledge:
Familiarity with MongoDB and React.js is a must.
Node.js is preferred for the backend, though you can choose other frameworks/programming languages if you are comfortable implementing the logic.

17. WebRTC Services

Mentors: Nikhil Gupta
Tech stack: FrontEnd: React / Next + tailwind / styled components, BackEnd: Firebase / Express,Database: Firestore / MongoDB

(These are what I would have preferred. You are not restricted to these. Feel free to suggest or use some other stack)

Difficulty: Medium/Hard

Description:
A web app which will leverage WebRTC technology to send files to a person or live stream video. Apart from creating the UI, this Frontend focused project will help you master handling large data files and use advanced browser APIs like user devices, service workers, etc. Overwhelmed? Don't be 😌. At the end, every complex application is just a beautiful arrangement of small and simple components.
Features:

- ➔ Dashboard
- ➔ Video streaming
- ➔ End-to-end encrypted networking
- ➔ Modern UI
- ➔ User Dashboard

Prerequisite knowledge:
Basic knowledge about Javascript / Typescript, Basic React, CSS, Express.

18. **Auth Microservice (Filled - Mentor cannot take anymore teams for this project)**
Mentors: Nikhil Gupta
Tech stack: Javascript / Typescript, ExpressJS, Docker (If you want to explore).
Difficulty: Medium/Hard

Description:
A backend project focused on authentication and authorization. You will be not just hashing passwords but will be implementing industry level auth flows. You will be interacting with async JWTs, access and refresh tokens, custom roles, etc. You will also be implementing MFA (using OTPs, Authenticator apps, etc.).
Features:

- ➔ Deployable as an independent microservice.
- ➔ Production ready backend architecture.

Prerequisite knowledge:
Basic knowledge about Javascript / Typescript and experience with creating basic CRUD APIs.

19. **Extension for productivity hacks in gmail**

Mentors: Nafees Nehar
Tech stack: javascript/typescript, basic web development
Difficulty: Medium/Hard

Description:
The goal is to get basic understanding about browser extensions by building a simple extension which will add an action button with every email in the gmail UI that will be used as a reminder like “remind me about this email in ____ time” just like how it happens in slack. The reminder would then be notified by the browser. This can be extended to attach personal notes with an email or watching the email for replies.

Prerequisite knowledge:
Javascript

App Development:

20. P2P Sharing PLatform

Mentors: Aaryak Shah, Harsh Sharma
Tech stack: Networking, App Development
Difficulty: Medium/Hard

Description:
Design and develop an application that can be used to transfer files between any two computers (peers) over a network. Refer to applications like DC++ and Torrents, and even Air Drop and Share It.
Features:
- ➔ A UI application that presents the functionalities
- ➔ Ability to download any "available" file from any active user on the network
- ➔ Ability to browse online users and their files available to be fetched
- ➔ Messaging/chatroom service for network users to discuss on

Non-essential, but Great Features:
- ➔ Pause/resume file downloads
- ➔ Ability to share entire folders instead of just files
- ➔ Network-wide search functionality

Prerequisite knowledge:

Any programming language. (Python is preferred and likely the easiest to work with due to its high level abstractions of various useful concepts)
Any UI framework. You may choose to create a browser application, similar to uTorrent Web, or you may use Qt or a similar library to build for desktop or even mobile. (Qt or a web framework is preferred as they are easy to use)

As you may have observed, this project focuses heavily on the design of the system architecture. To that end, we invite you to make various decisions regarding the architecture of your application. Whether or not it should be fully decentralized whether the application should work over a LAN or a WAN; what network technologies it should utilise, and other design questions are left open to discussion. You may design the system however you deem appropriate.

** Note that, we are not trying to create a cloud storage or file server. You are not expected to make a Google Drive-like service. Users will contribute files directly when they are connected. Other design choices are open to interpretation.

21. Android package (npm template)

Mentors: Manik Malhotra
Tech stack: FrontEnd: React / Next + tailwind / styled components,BackEnd: Firebase / Express,Database: Firestore / Mongo DB

(These are what I would have preferred. You are not restricted to these. Feel free to suggest or use some other stack)

Difficulty: Medium/Hard

Description:
It is a npm package for android generator basically it lacks a half component (the Java part is completed) and it needs a Kotlin section to fully make it complete. It provides a generator to create and maintain an Android application based on the latest frameworks and patterns used by the community.

Prerequisite knowledge:
JAVA, Data-Binding, MVVM-Architecture, Firebase, RxFirebase i.e: DAO

22. Android Application

Mentors: Anmol Srivastava

Tech stack: Simple Android Applications
Difficulty: Easy

Description:
Simple Android Application using Kotlin

Prerequisite knowledge:
Interest in Android Development

23. **Architectural Patterns based Applications**

Mentors: Anmol Srivastava
Tech stack: Android Applications
Difficulty: Medium/Hard

Description:
Android application using Kotlin, VM architecture, Coroutine, Team code architecture, Repository model

Prerequisite knowledge:
Interest in Android Development, Able to make basic android applications

Blockchain:

24. **Decentralized Twitter**

Mentors: Parth Lawania
Tech stack: Usually MERN but any changes are acceptable, IPFS, Smart Contracts(Solidity).
Difficulty: Medium/Hard

Description:
Create a decentralised twitter like application. The current tech stack includes the use of blockchain (smart contracts) and IPFS. The tech stack is totally flexible based on the comfort and architecture proposed.

Prerequisite knowledge:
Basic knowledge of full stack development is required.

25. Decentralized DigiLocker

Mentors: Kailash Kejriwal
Tech stack: Blockchain + Web Development
Difficulty: Medium/Hard

Description:
DocX is a decentralized document management system that keeps the tracks of the documents using the on-chain history and keeps the documents in encrypted form which provides safety and security.
Features:

- ➔ DocX allows the uers to request for the adminstrators of some selected departments to issue a new document or modify the existing documents.
- ➔ All of the actions are performed on-chain.
- ➔ This keeps the integrity of the entire process and prevents any third party influence.
- ➔ Users can also manage their documents in the application by downloading it and so on.

Prerequisite knowledge:
Fundamentals of Blockchain,Web Development

Cloud Computing:

26. Remote Code Executor

Mentors: Manish Kumar
Tech stack: Cloud - Virtual Machines, Orchestration Tools, Networking, Serverless Functions (AWS / GCP), Any Backend Framework (Preferably Node.js), HTML/CSS or Any Frontend Framework (Preferably ReactJS)
Difficulty: Medium/Hard

Description:
Build a Cloud based system to execute code snippets remotely on Virtual machine through the browser in multiple languages. Then incrementally upgrade it to make it scalable using Serverless tools or Orchestration tools.

Prerequisite knowledge:

Primary prerequisite is their willingness to learn, grind and commit. Some web development knowledge would be good, but make sure they know that it's not a requirement or necessity

Others:

27. **Local Dc++**

Mentors: Anurag Shrivastava
Tech stack: Computer Networking , Operating System .
Difficulty: Medium/Hard

Description:
A desktop application that runs on intra-network and replicates the features of DC++. The application uses various networking, multithreading concepts, and algorithms to achieve the best file transfer speed possible. This application will run on any local area network for chatting and file-sharing purposes. Downloading speed in Intra-Share will always be higher than one can get from torrent or any other website.

Prerequisite knowledge:
Basic knowledge of CN,Operating Systems