



# Guidelines and Problem Statements

## What is “WHAT THE HACK”?

**WHAT THE HACK** is a “**hackathon**” event conducted by the Technical Council at IITGN, where innovators and problem-solvers collaborate to tackle real-life challenges within a finite time frame. The goal is to create a solution from scratch and deliver a working prototype by the end. The most popular format is the weekend hackathon, which spans 24 to 36 hours of intense, focused development.

But there's more to this hackathon than just innovation—this event is also a gateway to joining the different Clubs of the Technical Council! We'll be scouting for the most promising talents throughout the hackathon, and the participants who impress us with their creativity, teamwork, and problem-solving skills will be inducted into our club. So, not only do you get the chance to build something amazing, but you could also secure your spot in the club, where the real journey of innovation and learning begins.

# Know Your Club Secretaries:

Contact the respective Secretaries for the respective problem statements.

Technical Club	Club Secretary	Contact Details
Metis, Development Club	Jayraj Dulange	9145609395
Digis, Digital Sports Club	Hari Balaji	8451814569
Mean Mechanics, Robotics Club	Malhar Karade	9209143733
Odyssey, Astronomy Club	Abdul Ronak, Abhay Upparwal 🌿	8120363851, 7060340966
GRASP, CP Club	Karan Gandhi, Nishchay Bhutoria 🌿	9920044269, 9353012199
Machine Learning Club	Shreyans Jain	9502685029
Systems Club	Mithil Pechimuthu, Tirth Patel 🌿	9901469198, 7622090375
Embed Club	Zaqi Momin	9998142340

🌿 : Co-secretaries

## FAQs About WHAT THE HACK:

**1. I've never participated in a hackathon before. Should I still consider taking part?**

**Answer** - Yes, and if you're interested - you must! You will learn a lot while working on the problems. You should make good use of all the available resources: Google, YouTube, GitHub, Stack Overflow and ChatGPT 😊 - but make sure you know and understand what the code is doing.

**2. Do I have to participate individually or do I have to form a team?**

**Answer** - Individual Participation only. (Discussions are allowed)

**3. Can I use AI tools and ChatGPT?**

**Answer** - Yes 😊

Hackathons are like a sprint, here, you should make use of all available resources. But we also want to induct members, hence, we request you to have a basic understanding of the code that you generate using AI tools. We also recommend using Google and Stack Overflow.

**4. How long is the hackathon?**

**Answer** - The hackathon starts at 4 pm, 31st August, and will be live till 11:59 pm, 1st September.

**5. Do I need to solve all the questions?**

**Answer** - We have several problem statements, you can solve as many as you want to. All problem statements will be judged individually based on their difficulty level, and each of them will have rewards for top submissions.

**6. Can anyone help me with doubts related to problem statements?**

**Answer** -

For further queries, you can reach out to the respective club secretaries or the organizers (the Technical Council). Feel free to drop a WhatsApp message or call on the numbers mentioned above.

We will have a doubt clearing + help session on 1st Sept, offline, from 2 pm onwards. Venue: AB 1/101, 1/102, 1/201. Here, you can show your work and discuss with the secretaries and the organizers.

**7. I forgot to register. Can I still participate?**

**Answer** - Yes, you can participate now. Fill out this registration form:

<https://docs.google.com/forms/d/e/1FAIpQLSdN-kR2ZHp6xjqlhEhAV5NCgHBFnVOxa4oHz6V-9Ua95raPgw/viewform>

**8. What results do I need to deliver/submit our work?**

**Answer** - At the end of the hackathon, you must submit your work via the submission link (will be shared soon!)

**9. What happens after the hackathon?**

**Answer** - Top submissions will get rewards. All submissions will be carefully judged by a panel composed of the respective club secretary and the organizers. Induction to various IITGN Technical clubs is also via this hackathon.

**10. What if I missed the submission deadline?**

**Answer** - Don't panic! You can reach out to respective club secretaries, or you can even contact our team, "Technical Council." We will always be there to help you in that situation. But still, try to be punctual about the deadlines.

## **FAQs About Clubs:**

Most common doubts are addressed in the following google doc, if you have more questions, feel free to reach out to the organizers or club secretaries.

[https://drive.google.com/file/d/1g-XDWAuu5YBg4q6v\\_L6KU2uh1jYlyZoK/view](https://drive.google.com/file/d/1g-XDWAuu5YBg4q6v_L6KU2uh1jYlyZoK/view)

# METIS - Development Club

## Elementary Track

### Problem Statement: Website and App Analysis for IITGN, IMDb, and insIIT

This task is all about exploring and analyzing two popular websites, [iitgn.ac.in](http://iitgn.ac.in) and [imdb.com](http://imdb.com), and one application, **insIIT**. You'll get to dive into their user experience (UX) and user interface (UI), examining what works well and what could use some tweaking. You'll also take a peek behind the scenes to see what technology powers these platforms. The goal is to build your analytical skills while having a bit of fun exploring how different digital platforms are put together!

### Suggested Features

1. **Performance Analysis:** Start by analyzing how each platform performs. Look at things like how fast the pages load, how responsive they are, and how smooth the overall user experience feels. Does it lag?
2. **UX Evaluation:** Dig into the user experience of each platform. Is it easy to navigate? Are things where you expect them to be? How accessible is it for all users? Point out what you think works well and what doesn't, and don't be shy about giving your honest feedback.
3. **Tech Stack Analysis:** Get curious about the tech behind the scenes. What backend technologies are these platforms using? Does the tech stack fit the platform's needs? Explain why or why not, with your own take on whether it's the right choice for each. You can use [this](#) tool for analyzing the tech stack.
4. **Endpoints:** Explore different endpoints provided by the site and their uses. What underlying technologies are they using to create these?

### Evaluation Criteria:

- **Thoroughness:** How well did you cover each part of the task? Did you dig deep into performance, UX, UI, and tech stack? Your analysis should be detailed and cover all the basics.
- **Insightful Feedback:** It's not just about spotting flaws—offer genuine insights and practical suggestions for improvements. Show that you understand not just what's wrong, but how it could be better.
- **Clarity and Justification:** Your reasoning should be clear. When you like or dislike something, explain why. The more you justify your claims with solid reasons, the stronger your report will be.

## **Problem Statement: To-Do App**

To-Do App is a simple tool designed to help students keep track of their daily tasks and assignments. The idea is to make a straightforward app where users can quickly add tasks, mark them as done, and see what's coming up next. The notes should be saved permanently, so they don't get lost even if the app is closed or restarted.

### **Suggested Features:**

1. **Add Tasks:**
  - Users should be able to add new tasks with a name and an optional due date.
2. **Task List:**
  - A main screen that displays all tasks in a clear list, showing which ones are pending and which are completed.
3. **Mark as Complete:**
  - A simple way to mark tasks as done, which will then move them to a completed section.
4. **Edit and Delete:**
  - Option to edit task details or delete tasks.
5. **Reminders:**
  - Basic reminders for tasks with due dates, so users get a notification as the deadline approaches.

### **Evaluation Criteria:**

- **Depth:** The app should be straightforward and intuitive. Users should be able to figure out how to use it without any help.
  - **Functionality:** It is expected that all the basic features listed above should work reliably without errors.
  - **Design:** The design should be clean and not too cluttered. It should be easy on the eyes, with a simple color scheme and readable fonts.
- 

## **Intermediate Track**

### **Problem Statement: Web-Based Chat Application**

The aim of this problem statement is to build a web-based real-time chat application that allows users to send and receive messages instantly. The app should support individual chat rooms and group messaging. Users will be able to log in, search for other users, and communicate

seamlessly with both individuals and groups. The focus should be on building a smooth, responsive, and scalable messaging platform using standard web technologies.

### **Suggested Features:**

1. **User Authentication:**
  - Implement a secure user authentication system, allowing users to sign up, log in, and log out.
2. **Real-Time Messaging:**
  - Enable real-time messaging using WebSockets or Firebase. Messages should appear instantly in the chat window without needing to refresh the page.
3. **User Search and Contact List:**
  - Allow users to search for other registered users by name or username.
  - Provide a contact list for users to easily view and select contacts to start a chat.
4. **Group Chat Functionality:**
  - Allow users to create group chats, add/remove members, and send group messages.
  - Display the list of group members and notify everyone in the group when a new message is sent.
5. **Responsive UI Design:**
  - Ensure that the chat application works smoothly on both desktop and mobile devices, with a responsive design that adjusts to different screen sizes.

### **Evaluation Criteria:**

- **Functionality:** The core messaging system must work smoothly, allowing real-time text-based communication. Features like user authentication, search, and notifications should function without glitches.
- **User Experience:** The UI/UX design should be intuitive and user-friendly, with a clean layout and easy navigation. The chat interface should be responsive and accessible from different devices (desktop, tablet, mobile).
- **Performance:** The application should handle real-time communication efficiently, with minimal latency. It should support multiple users without significant slowdown.

---

## **Advanced Track**

### **Problem Statement:** Resume Builder using OpenAI/Groq AI API (Web Development)

The goal of this project is to build a web-based application that helps users create resumes quickly by using the OpenAI API/Groq AI API with minimal effort. Users will input their personal and professional details, and the app will automatically generate a well-structured, polished

resume in a professional format. The idea is to simplify the process of creating a resume by providing suggestions and formatting tips.

### **Suggested Features:**

1. **User Input Form:**
  - Provide a form where users can enter details like name, contact information, education, work experience, and skills.
  - Ensure validation for key fields (e.g., required fields, email format).
2. **AI-Powered Content Suggestions:**
  - Use the OpenAI API to generate content, such as job descriptions, skill sets, and accomplishments based on user input.
  - The AI should suggest action-oriented bullet points and impactful phrasing for different sections of the resume.
3. **System prompt for content generation:**
  - Implement a **system prompt** when making API calls to ensure that the generated resume content adheres to a professional tone and style. This prompt should guide the AI to produce concise, formal, and well-organized resume entries.
  - Example system prompt: *"You are helping users write professional resumes. Provide clear, action-oriented suggestions for job descriptions, skills, and accomplishments. Use a formal tone and keep sentences concise."*
4. **Resume Formatting Options:**
  - Allow users to select from different layout templates (e.g., modern, minimalist, professional) that are well-suited to various industries.
5. **Live Preview:**
  - Show a real-time preview of the resume as the user fills out the form.
6. **Additional Features:**
  - **LaTeX Format Support:** Provide a LaTeX template option for users who prefer high precision and control over formatting. Generate a downloadable LaTeX file with auto-filled content. ***Bonus for including CDS format.***
  - **Edit Suggestions:** Offer suggestions for improving the resume, such as highlighting key skills or rewording sections for better impact.
  - **You can add other features also which can help users to make their resume.**

### **Evaluation Criteria:**

- **Functionality:** The website should perform the core tasks effectively—taking user input, generating content with the OpenAI API, and creating both PDF and LaTeX resumes.
- **Ease of Use:** The application should have a simple and intuitive user interface. Users shouldn't struggle to input their data or download the resume in both formats.
- **Content Quality:** The AI-generated suggestions should be relevant and well-suited to the user's input, making the resume content stand out in a professional way.

- **API Usage:** Ensure the efficient use of the OpenAI API, minimizing token usage while still generating high-quality content.
- 

### **Development Resources:**

You are also *free to use any tech-stack of your choice* apart from the below mentioned:

- [NodeJS](#), [Django](#), [PHP](#), [MongoDB](#), [MySQL](#), [FastAPI](#), [Postgres](#) (backend & database)
- [ReactJS](#), [AngularJS](#), [Bootstrap](#), [HTML](#), [CSS](#) (frontend)
- [Google Firebase](#) (All in one service for faster development)
- Frontend Deployment: <https://vercel.com/docs/deployments/overview>
- Server Deployment: <https://docs.render.com/free>

### **Submission Guidelines:**

You will be asked to submit the following:

- Link to your deployed application.
- Link to your GitHub repository that contains all the code/report. Ensure that the visibility of the repository is set to public.



# DIGIS - The Game Development Club

## Introduction

Welcome to DIGIS, a place where people celebrate their love of video games and tinker with different ways to make games and enjoy them together.

## Recommended Resources

Godot Engine: <https://godotengine.org/>

Open Source Assets:

<https://opengameart.org/>

<https://itch.io/game-assets/tag-open-source>

<https://kenney.nl/>

<https://mixkit.co/>

Game Development Subreddit: <https://www.reddit.com/r/gamedev/>

Beginner Videos:

<https://www.youtube.com/watch?v=L0hfqjmasi0>

<https://www.youtube.com/watch?v=e1zJS31tr88>

## Problem Statements

***While you only have to choose any one of the following four problem statements, you are encouraged to attempt multiple. Your submission will be judged accordingly.***

**Simple:**

### 1. Stackability Theme

**Problem Statement:** The challenge is to create a game that incorporates the concept of “stackability” as a core mechanic. The game should revolve around stacking, layering, or balancing elements in an engaging and fun way. Whether it’s objects, game mechanics, levels, or even narratives, the concept of stacking should be central to the gameplay experience. Your goal is to explore creative interpretations of what it means to “stack” and how this can be turned into an interesting game mechanic.

**Some Guidelines:**

1. **Keep It Simple:** Start with a simple idea that fits the theme and focus on executing it well.

Don’t try to do too much in a short time.

2. **Define Your Core Mechanic:** Decide early on what kind of stacking mechanic you want to implement. Is it stacking objects, building towers, combining abilities, or something else? Stick to one core idea.

3. **Prototype Early:** Build a quick prototype to test your core mechanic. This will help you identify what works and what doesn’t early on, so you can iterate.

4. **Focus on Gameplay:** While visuals and sound are important, focus on making your gameplay fun. You can also add small unique mechanics which make the game more interesting.
5. **Manage Scope:** Keep your project manageable. Prioritise the most essential features and avoid adding too many complex elements.
6. **Get Feedback:** Share your game early with others to gather feedback. This will help you identify areas for improvement and new ideas.
7. **Have Fun and Be Creative:** Game jams are all about experimentation and learning. Don't stress too much about perfection; focus on being creative and having fun!

Remember, your interpretation of “stackability” can be literal or abstract, as long as it aligns with the theme in a meaningful way.

**Recommended Game Engine: Godot Engine**, it is an easy to use open source game engine. You do not have to use the recommended game engine. You can use any game engine which you feel comfortable with.

---

## 2. Survival Theme: Stay Alive as Long as Possible

**Problem Statement:** The goal is to create a game where the primary objective is to survive for as long as possible. The player should face increasing challenges, limited resources, or dangerous environments that require strategic decisions and quick reflexes to stay alive. The game can be set in any context—whether it's surviving in a post-apocalyptic world, outlasting waves of enemies, managing dwindling resources, or avoiding hazards in a harsh environment. The focus is on creating a tense and engaging experience where survival is constantly at risk.

### **Some Guidelines:**

1. **Choose Your Survival Context:** Decide the environment or scenario where the player needs to survive. It could be a wilderness survival game, a dungeon crawler, a zombie apocalypse, or even a space adventure.
2. **Define Core Mechanics:** What will challenge the player's survival? Consider key survival elements like managing resources (food, water, oxygen), battling hostile forces, enduring environmental dangers, or avoiding traps. Pick one or two core challenges to focus on.
3. **Create a Sense of Urgency:** Include time-sensitive challenges, limited supplies, or growing threats that force the player to think quickly and make tough decisions.
4. **Simple, But Engaging:** Keep your design focused. Make sure the gameplay loop is simple but fun, like collecting resources, crafting, or defending against waves of enemies.
5. **Balancing the Challenge:** Test your game frequently to ensure that it's challenging without being impossible. Introduce difficulty gradually to give players time to learn and adapt.

The "Stay Alive as Long as Possible" theme is versatile, so experiment with unconventional ideas within the survival genre. The key is to create an experience where players are constantly challenged, adapting, and striving to survive just a little longer.

---

## **Advanced**

### **1. Two-Button Game:**

**Problem Statement:** The challenge is to design a game that can be entirely played using only two buttons. With such a strict limitation, creativity and clever design are key. Despite the simplicity, your game should be engaging, challenging, and fun. Whether it's a platformer, puzzle, rhythm game, or something completely different, the objective is to show how much can be achieved with just two inputs. How can you create depth, strategy, or excitement with such minimal controls?

#### **Some Guidelines:**

- 1. Choose an Intuitive Control Scheme:** Decide how the two buttons will function. Common setups could be left/right, jump/dash, or a combination of actions based on timing or context. Keep the controls simple and easy to grasp.
- 2. Design Around Simplicity:** Since you only have two inputs, the gameplay should focus on how these buttons interact. Consider mechanics like timed presses, holding vs. tapping, or alternating between actions. The simpler the mechanics, the more accessible your game will be.
- 3. Pick a Genre that Fits:** Certain genres work especially well with limited controls, like endless runners, rhythm games, and puzzle games. Choose a genre that naturally aligns with a minimal control scheme.
- 4. Focus on Gameplay Depth:** Depth doesn't have to come from complex controls. Introduce variety through level design, increasing difficulty, or evolving challenges. Make the gameplay feel dynamic even with limited input options.
- 5. Polish and Feedback:** With such a focused design, small details matter. Make sure the controls feel responsive, the challenge is fair, and the game is polished. Clear feedback for player actions (like visual or sound cues) can enhance the experience.
- 6. Short Play Sessions:** Games with limited controls often shine in short bursts. Aim for quick, replayable experiences that players can enjoy repeatedly without it feeling repetitive.
- 7. Experiment with Timing and Precision:** Since you have only two buttons, consider using timing and precision as core mechanics. Games like "Flappy Bird" or rhythm games thrive on well-timed inputs that can keep players engaged.

This theme is all about embracing constraints and finding creative ways to do more with less. With only two buttons, the challenge lies in turning simple mechanics into engaging, fun experiences.

---

## 2. Basic Platformer:

**Problem Statement:** Create a straightforward platformer game featuring at least two levels and a boss fight. The player should navigate through obstacles, defeat enemies, and reach the end goal in each level. The game should have a gradual difficulty increase, with the first level introducing core mechanics, the second level raising the challenge, and a final boss fight to wrap up the experience.

### Some Guidelines:

1. **Core Mechanics:** Focus on basic platforming actions like running, jumping, and potentially attacking. Ensure these core mechanics feel responsive and fun.

### 2. Level Design:

Level 1: Introduce mechanics in a simple environment that helps the player learn.

Level 2: Add more difficult jumps, obstacles, or enemies to challenge the player.

3. **Boss Fight:** Design a boss with clear attack patterns and weak points. The fight should test the player's mastery of the skills learned in the previous levels.

### 4. Focus on Flow and Pacing:

Keep the game's difficulty gradual, and ensure the player's progression feels smooth.

### 5. Polish the Basics:

Make sure sound effects, visual feedback, and controls feel polished to enhance the overall game experience.

By focusing on tight controls, good level design, and a balanced boss fight, you can create a fun and engaging platformer that's easy to complete within the jam's timeframe.

### Other Rules:

- Usage of open source assets is encouraged
- There are bonus points for adding unique mechanics other than the game's theme
- **The time limit is the duration of the hackathon** (from Saturday 14:00 to Sunday 23:59)

## Selection Criteria

- Playability of the game
- Overall aesthetic and design of the game (including the sound)
- Unique mechanics and features (not in quantity but quality)

## Submission Guidelines

Submission guidelines will be notified soon, right here. Stay tuned!

# Mean Mechanics, the Robotics Club

## Yo peeps !

As interesting as the word “robotics” sounds, trust the facts, mate, that it’s pretty engaging & tiring to it’s equal.

As a part of the club some of its members have demonstrated extensive work & hard work towards the subject. It has produced many productive results, such as qualification of [Robofest 4.0 round 1](#)

If you also wanna join the club to say anything, deepen your knowledge, build interesting stuff only because it's “interesting”, having fun or whatever, here’s your chance to prove yourself worthy of the same.



As a member of the club you can join as a core-member or as a mentee. As the words suggest, the core-members will be the ones theoretically “senior” to the mentees. However, this is simply a setup for functioning of the club and as it’s a topic that merits more discussion, we’ll simply say that both people will form the essential pillars of Mean Mechanics....

## Problem statements

- [Problem statements for mentees](#)
- [Problem statement for core members](#)

## Note:

- 1) Both the documents contain a basket of problem statements. You can choose any of them, but your work, understanding & explanation should be thorough & concrete.
- 2) This round will serve as “Qualifier 1” & will be followed by an interview.
- 3) Issue the electronics only after your mechanical setup is right as rain
  - Contacts for Electronics :
    - Malhar : 9209143733
    - Zaqi : 9998142340

# GRASP, CP Club

## Introduction

Welcome to the Competitive Programming Club!

Get ready to sharpen your problem-solving skills, tackle complex algorithmic challenges, and compete with some of the brightest minds. Whether you're optimizing your code for runtime, reducing space complexity, or finding elegant solutions to tricky problems, this is where you push your limits and grow as a coder!

## Fun Fact About Competitive Programming

Did you know mastering competitive programming can give you a significant edge in job interviews and coding challenges? Here's why:

- **Algorithmic Aces:** Those who excel in competitive programming are often well-versed in data structures and algorithms, making them formidable candidates in technical interviews.
- **Speed Coders:** Writing efficient code quickly is a skill that top tech companies highly value.
- **Puzzle Masters:** Competitive programmers are known for their ability to think outside the box and solve problems under pressure—skills that translate well to any technical role.

## About Competitive Programming Club

The Competitive Programming Club is a hub for enthusiasts, students, and professionals who love the thrill of coding contests. If you're passionate about algorithms, enjoy solving puzzles, or want to prepare for programming competitions like ACM-ICPC, Codeforces, or LeetCode, this is the place to be!

## Core-Members Induction:

You can become a core member of the club in the following ways:

- If your rating on Codeforces is above 1600 or that on CodeChef is above 1900. You can directly be inducted as a core member through an interview.
- If you don't meet the above criteria, don't worry; you can give our core-member induction contest, a long competitive programming contest hosted on Codeforces.
- Here's the link to the contest:  
<https://codeforces.com/contestInvitation/7f840817a6a9e253b89c94f03af428e3750dca5c>

## Mentees Induction:

You can become a mentee of the club in one of the following ways:

- We'll be releasing a series of puzzles for you to solve, where you'll need to explain the reasoning behind your logic. Your selection as a mentee of the club will be based on how many questions you get right.
- If you're already experienced in competitive programming, you can participate in the same contest as our core members, and your selection will be based on your performance. You may be asked to appear for an interview in addition to your contest performance. This is to combat cheating.

Here is the link to the Mentees induction contests: [📄 GRASP Mentee Induction Contest](#)

# Machine Learning Club

## Hey Everyone!

Welcome to the Machine Learning Club's Induction Hackathon!

Machine learning is all about curiosity, innovation, and pushing boundaries. At the ML Club, we're building a community where anyone—whether you're just starting out or already experienced—can come together to learn, experiment, and create amazing projects. This isn't just another club; it's a place where ideas turn into action, and learning happens by doing.

## Why join us?

- **Real-World Impact:** Work on projects that matter. From collaborations with industry experts (yes you get paid to learn!) to research that contributes to the ML community, we focus on initiatives that make a difference.
- **Learn by Doing:** Forget boring lectures. Here, we dive straight into hands-on workshops, hackathons, and projects where you learn the latest in ML by actually building and experimenting.
- **Inclusive for All:** Whether you're a newbie eager to learn or a seasoned coder looking for your next challenge, we have something for you. Our activities are designed to be inclusive, challenging, and fun for all skill levels.

## About the Hackathon:

Today's hackathon is your chance to dive in and experience what the ML Club is all about. We've got some exciting problem statements lined up that will challenge you to think creatively, code smartly, and collaborate effectively. This is more than just a competition; it's a chance to learn, grow, and maybe even spark your next big idea!

So, let's get started and have some fun. Show us what you've got!

## Resources:

*Check out these essential [resources](#) to kickstart your ML journey.*

**Note:** We take plagiarism seriously and do not want you to copy directly from the internet and pass it off as your own. You are allowed to use code from the internet, but it's crucial to understand the intent behind using it. You are encouraged to use large language models like **ChatGPT**, but ensure you comprehend what you are doing. Be prepared, as a random subset of participants might be called for interviews.



## Beginner Track

**Problem Statement:** Explore a dataset, frame interesting questions, and gain insights through basic data analysis techniques.

### Tasks:

#### 1. Dataset Exploration and Preparation:

- Load the provided [dataset](#) using Python libraries such as Pandas and NumPy.
- Perform Exploratory Data Analysis (EDA):
  - Understand the structure of the dataset (e.g., rows, columns, data types).
  - Visualize data distributions using basic plots (histograms, bar charts).
  - Identify missing values or outliers and discuss potential impacts on your analysis.

#### 2. Framing Questions and Gaining Insights:

- Develop interesting questions you want to answer using the dataset. For example:
  - What factors influence a particular outcome or variable?
  - Are there patterns or trends in the data over time?
  - How do different variables relate to each other?
- Use your questions to guide your analysis and identify key insights from the data.

#### 3. Data Cleaning:

- Address missing values using methods like imputation or removal.
- Standardize or normalize data to make it suitable for analysis.
- Convert categorical variables into numerical formats if necessary (e.g., using one-hot encoding).

#### 4. Basic Analysis and Insights:

- Calculate summary statistics (mean, median, mode, etc.) to understand the data's central tendencies.
- Use Pandas and NumPy for basic data manipulation and answering your framed questions.
- Create visualizations (scatter plots, line graphs) to explore relationships and trends in the data.

#### 5. Introduction to Predictive Modeling:

- Split the dataset into training and testing sets.
- Apply simple predictive algorithms such as Linear Regression or Decision Trees.
- Evaluate model performance using basic metrics (accuracy, MAE, etc.).
- Relate your model's findings back to the questions you framed.

### Evaluation Criteria:

- **Question Framing and Insight Development (30 points):** How well did you frame questions and gain insights from your analysis?
- **Understanding of EDA and Data Cleaning (30 points):** How effectively did you explore and clean the dataset?

- **Application of Basic Analysis Techniques (20 points):** Were you able to analyze the data to answer your questions?
- **Introduction to Modeling (10 points):** Did you apply any predictive algorithms, and how did they contribute to answering your questions?
- **Presentation and Documentation (10 points):** Quality of your code, documentation, and explanation of your findings and process.

**Bonus (10 points):** Go further by exploring additional algorithms or techniques, or by developing more advanced visualizations to enhance your insights.

---

## Intermediate Track

**Problem Statement:** Develop a neural network that learns to apply the Sobel filter to images.

### Tasks:

1. **Dataset Preparation:**
  - Use the provided [dataset](#) of images.
  - Apply the [Sobel filter](#) to the dataset using a standard image processing library.
  - Save the original and Sobel-filtered image pairs.
2. **Model Development:**
  - Design a neural network that takes an original image as input and produces a Sobel-filtered image as output.
3. **Training:**
  - Train your model using the prepared dataset.
4. **Evaluation:**
  - Evaluate your model's performance on a provided test set.

### Evaluation Criteria:

- **Model Accuracy (50 points):** How well does your model replicate the Sobel-filtered images?
- **Code Quality (20 points):** Is your code well-organized, documented, and efficient?
- **Innovation (15 points):** Any creative or unique approaches to solving the problem?
- **Presentation (15 points):** Quality of your final presentation, clarity of explanations, and insights derived from your results.
- **Bonus (15 points):** Extend your project by applying the same approach to other classical image filters such as the Laplacian Filter and Prewitt Filter. Additionally, visualize and compare representations of different CNN layers as images. Explore various CNN architectures to gain deeper insights into their effects and performance. This will demonstrate your ability to generalize and analyze the model's behavior across different scenarios.

## Advanced Track

**Problem Statement:** Develop an autoencoder network to add colour to grayscale images using self-supervised learning techniques.

### Tasks:

1. **Dataset Creation:**
  - Create a diverse dataset of colour images, ensuring variety and representation of campus environments.
  - Implement data augmentation techniques to expand your dataset.
2. **Autoencoder Development:**
  - Design an autoencoder architecture suitable for the colourisation task.
  - Develop a self-supervised learning approach to train your model without explicit colour labels.
3. **Training and Colorization:**
  - Train your model to reconstruct full-color images from grayscale inputs.
  - Evaluate your model's performance on a hidden test dataset provided by the organizers.
4. **Additional Features:**
  - **Image Inpainting:** Add a feature to your model to reconstruct missing or corrupted parts of an image. Test on images with various types of degradation (e.g., scratches, missing patches).
  - **Resolution Enhancement:** Implement a super-resolution module to upscale low-resolution grayscale images. The model should both colourise and enhance the resolution in a single pass.

### Evaluation Criteria:

- **Accuracy of Colorization (55 points):** How does your model colourise the hidden test dataset accurately?
- **Dataset Diversity and Representation (15 points):** How diverse and representative is your created dataset?
- **Innovation and Creativity (10 points):** Any novel techniques or ideas implemented in your approach?
- **Code Quality (10 points):** Is your code clear, concise, and well-documented?
- **Presentation (10 points):** Quality of your final presentation, clarity of explanations, and insights derived from your results.

**Bonus Feature Integration (15 points each):** How well does your model integrate the additional features (image inpainting and super-resolution)?

**Submission Details:** Details for submission shall be provided here soon. Keep your Python files, Jupyter notebooks and media (if any) ready!

# Systems Club

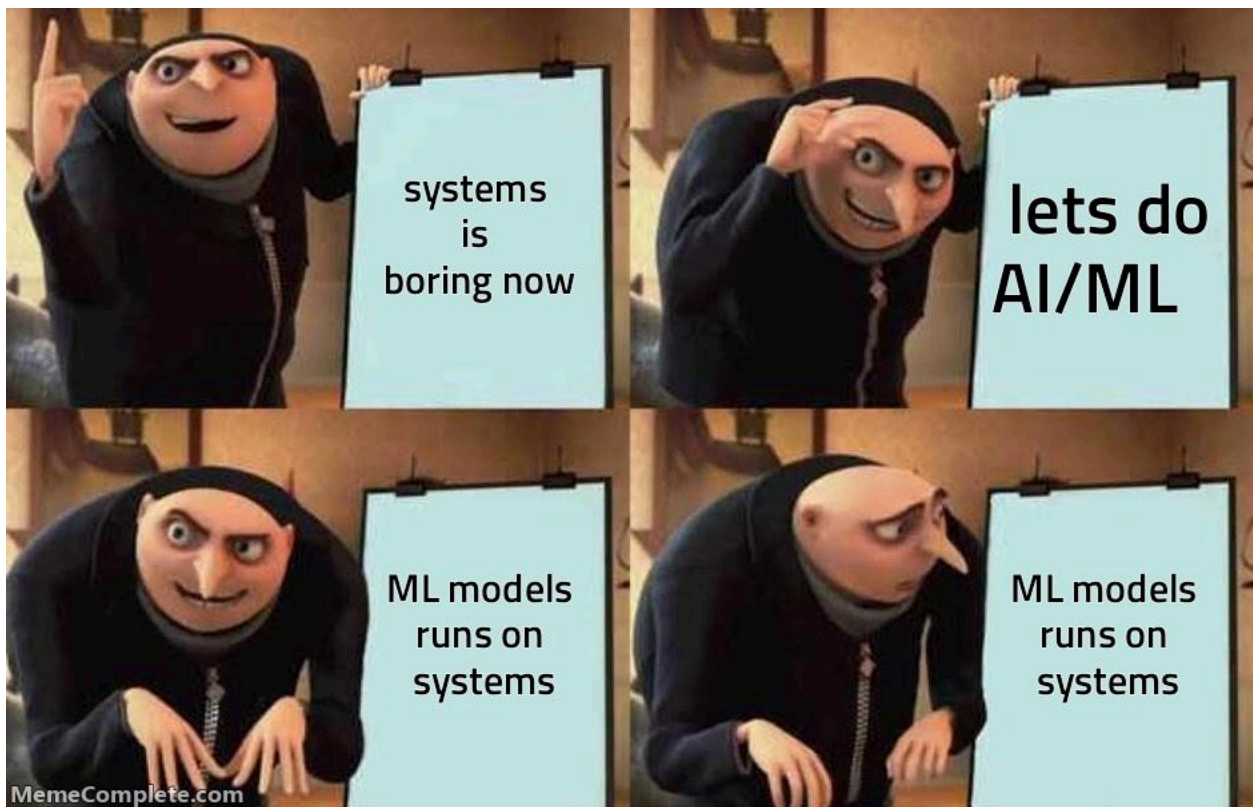
- > Why do Linux programmers prefer to use the command line?
- > Because they can't C without it, Duh! 😊

## Introduction

Welcome to the Systems Club Hackathon!

Ready to flex your hacking skills, secure some systems, and show off your programming chops? Whether you're decrypting messages, exploiting binaries, or debugging network configs, this is your chance to shine and learn something new!

## Fun Fact About Salaries!



Did you know that systems folks often have the edge when it comes to making the big bucks? Here's the scoop:

- **Systems Wizards:** Tend to bag more gold than others. Why? Because knowing how to tweak those kernels and secure those networks isn't just cool—it's lucrative!

- **Software Sorcerers:** They do pretty well too, especially when they conjure up some neat apps. But sometimes, their wallets aren't as full as our system pals'.
- **AI Alchemists:** Sure, they're the talk of the town right now, but their spells (or machine learning models) require a ton of training. And advanced degrees can cost a fortune!

## About Systems Club

Systems club is for amateurs, passionate students and professionals focused on exploring the world of systems programming, cybersecurity, and networking. If you love diving deep into the nuts and bolts of how things work, or if you're eager to learn, this is the place for you!

## Guidelines

1. **Prioritize Official Resources:** Always try to find solutions using official documentation or trusted online resources first. Reading man pages is really fun 😊.
2. **Use LLMs as a Last Resort:** While Large Language Models are cool, try to think through the problem yourself first! LLM - Looser's Like Me
3. **Explain Your Submission:** Be prepared to explain every aspect of your solution—no black-box magic!
4. **Environment Matters:** Use a Linux-based OS or WSL2 for Windows to ensure you're all set for the tasks ahead. Keep it ready before tomorrow.
5. **Stay Ethical:** This is all about learning and growing. Play fair, and have fun!
6. **If in doubt, contact us.**
7. **Submission platform will be out soon.**

## Problem Statements

### Logging/System Stats/Data Backup

If `x` is the exit status when nothing was found that matched the criteria specified to `apropos`.

`YOUR FLAG IS with{x}`

You are required to write a Linux shell script that performs two tasks:

#### Task 1: Log Generation

- Write a shell script that generates logs at 1 second intervals that store the timestamp, cpu load, memory consumption percentage, task counter value, sum of all PIDs and cpu temperature.
- This log should be stored in a file called ``system.log``.
- Use of man pages is highly encouraged to find suitable flags for commands.
- `man <app>`: to get man page of the `<app>`
- `apropos <description>`: to get list of applications capable of `<description>`

#### Task 2: Log Backup

- Create another script that continuously monitors the `system.log` file and take backup of the `system.log`.
- This script must accept a flag to tell if the script is being run in the test or production environment. In the test environment, the backup must be taken every 10 seconds. Whereas in the production environment, the backup must be taken every 5 minutes.
- `<your-script> --prod` or `<your-script> --test`
- Backup them as a gzip archive file with a filename:  
`bak-system-<log-start-timestamp>-<log-end-timestamp>.tar.gz` and save it locally.
- The complete log should be presented such that there must be no duplicate timestamps and no logged data must be lost.

### Submission Requirements

- Screenshots of the resources you searched to get the timestamp, CPU load, memory consumption percentage, task counter value, sum of all PIDs and CPU temperature must be submitted. Please put them in a document.
- Submit the scripts that you built.

### Evaluation Criteria

- You will be assessed based on the level of exploration of resources and the quality of your script.

## Network Configuration And Debugging

### Task

- Download the [`break`](#) binary file. This file will break your network configuration. You need to fix it by writing a shell script.

### Submission Requirements

- Submit a report describing the issues you found and how you debugged and fixed it.
- Submit the fix shell script.

### Evaluation Criteria

- You will be assessed based on the level of exploration of resources and the quality of your script.

**Note:** If you were unable to solve this, use the [`fix`](#) binary to fix the issue and continue with the other questions.

## Parallel Programming

If  $x$  is the year OpenMP was released for Fortran 1.0 and  $y$  is the minimum number of attributes in the `MPI_STATUS` structure,

YOUR FLAG IS `wth{x_y}`

### Task 1: Develop Gene Sequence (Credits: Professor Sameer G Kulkarni)

- Develop a program that generates a sorted list of all possible DNA sequences composed of the bases A, C, T, and G, up to a specified length. For example, sequences including "A", "C", "T", "G", "AA", "AC", ..., "TTTT", etc are expected when the length required is 4. The generated sequences must be saved to a file. You may use OpenMP, MPI, Bend (programming language), unix processes, pthreads or any other similar construct in any language you like.
- Note: The maximum length ( $n$ ) should be taken as input.  $1 \leq n \leq 10$

### Submission Requirements

- Submit the source code that solves the task.
- Explain your program logic in simple words.

### Evaluation Criteria

- The submission with the least runtime that generated the correct sequence will be awarded greater points.
- The program's quality will also be judged.

### Task 2: Rebuild Logs

- Using the backups you generated in the log generation task, develop a program to combine and reconstruct the original log. Make sure that the order of logs are coherent. You may use OpenMP, MPI, Bend (programming language), unix processes, pthreads or any other similar construct in any language you like.

### Submission Requirements

- Submit the source code that solves the task.
- Explain your program logic in simple words.

### Evaluation Criteria

- The submission with the least runtime that generated the correct logs will be awarded greater points.
- The program's quality will also be judged.

---

## CTF

We are currently adding more CTF problems. Please follow the link below to play. If you have time to spare, read through C programming vulnerabilities, Steganography, Packet Capture and

replay, Linux fundamentals, Common web server vulnerabilities, Cryptography.

**CTF URL:** <http://10.7.35.138:8787/>



# Embed, The Embedded Systems Club

The Embedded Systems Club is all about creating, exploring, and innovating with hardware. From microcontrollers to FPGAs, real-time operating systems to bare-metal programming, this is the perfect place for you if you're passionate about making things work at the hardware level. Whether you're a beginner or a seasoned tinkerer, you'll find a home here.

We regularly organise hands-on workshops where you can learn and apply your skills to exciting projects. In collaboration with Tinkers Lab and Maker Bhavan, you'll have access to state-of-the-art tools and resources to bring your ideas to life. Whether building intelligent IoT devices, crafting custom hardware solutions, or experimenting with microcontrollers, we empower you to make cool stuff and push the limits of innovation.

Our partnerships with the IRP Council allow members to work on real-world projects that can have a broader impact within and beyond IITGN. You'll gain technical skills and be part of a dynamic community that loves exploring the cutting edge of embedded systems and hardware design.

## Problem Statements:

<https://axiomatic-building-edb.notion.site/Embedded-Systems-12596842677248369dd3e187806c33e7?pvs=4>

Software engineers: it's just too expensive to optimize our code, also computers will be faster anyway in a few years

Hardware engineers:



## Hardware Eng



I built a CISC microprocessor out of sand and molten lead

## Comp Sci



My Python script isn't compiling in my Chrome IDE  
:(

# Odyssey, The Astronomy Club

Have you ever wondered what the rings of Saturn and the moons of Jupiter look like up close? If so, we invite you to join us at Odyssey, the astronomy club of IIT Gandhinagar. At Odyssey, we delve into a wide array of fascinating topics, from observing deep sky objects to making thrilling visits to ISRO labs and exploring the intriguing field of astrophysics. Come be a part of our journey and experience the wonders of the universe firsthand!

## Why join us?

1. **Explore Deep Sky Objects:** You'll have the unique opportunity to observe stunning deep sky objects such as galaxies, open star clusters, and nebulae using our state-of-the-art telescopes. Not only will you get to see these celestial wonders up close, but you'll also learn how to operate a variety of different telescopes, gaining valuable hands-on experience.
2. **Analyze with Cutting-Edge Technology:** With our Unistellar eVscope 2, you can delve into detailed analysis of specific deep sky objects. This advanced telescope provides you with high-quality data that enhances your understanding of the universe.
3. **Learn Astrophotography:** If you're interested in capturing the beauty of the night sky, you'll have the chance to learn and practice astrophotography. Odyssey also participates in various competitions, offering you the platform to showcase your skills and compete with fellow enthusiasts.
4. **Exclusive Visits to ISRO and SAC Labs:** As a member of Odyssey, you will have the exceptional opportunity to visit prestigious institutions such as the Indian Space Research Organisation (ISRO) and the Space Applications Centre (SAC). These visits will give you an insider's view of cutting-edge space research and technology, allowing you to observe firsthand the work being done in space exploration and satellite applications. This unique experience will provide you with invaluable insights into the operations and innovations of these leading space agencies.
5. **Experience College-Funded Trips:** Odyssey is renowned for organizing college-funded trips to exciting locations such as Mount Abu, Mandi, and IUCAA Pune. These trips provide you with the opportunity to experience different astronomical environments and gain broader insights into the field of astronomy.

## How to join us?

The process for joining Odyssey is a bit different compared to other clubs. Over the next few weeks, we will be organizing a series of events designed to introduce you to the fascinating field of astronomy. These events will provide you with an opportunity to explore various aspects of astronomy and engage with our activities.

Following this initial period, we will evaluate potential new members based on their participation and enthusiasm during these events. After a month, we will select our core members through an induction process. This selection will be based on your demonstrated interest and active involvement in our activities, ensuring that those who join are truly passionate about astronomy and committed to our club's objectives.