Task 2 : Password Cracking with Hashcat

Objective: **Crack password hashes using dictionary and brute-force attacks with tools like Hashcat.**
Tools : **Hashcat, John the Ripper, Kali Linux, rockyou.txt**
Deliverables: Hash list, cracked output, explanation of attack strategy

Hints/Mini Guide:

Start with simple MD5 hashes. Use --help to understand Hashcat modes. Try dictionary attacks first.

Outcome: Learn how weak passwords are cracked and how to defend against such attacks.

Interview Questions:

1. What is a hashing algorithm?
2. How does brute-force password cracking work?
3. What is the difference between hashing and encryption?
4. What is salting in password hashing?
5. What is a rainbow table?
6. Which tool is better: Hashcat or John the Ripper?
7. What is a dictionary attack?
8. Why are short passwords insecure?
9. What is the function of rockyou.txt?
10. How do attackers obtain hashed passwords?

Key Concepts: **Hashing, Brute-force, Dictionary Attack, Password Security, Rainbow Tables**

🌟 Task Submission Guidelines

🕙 Time Window:

You've got a 12-hour window—from 10:00 AM to 10:00 PM—to give your best. It's your time to shine. But remember, once the clock hits 10:00 PM, submissions close.

🔍 Self-Research Allowed:

You're not alone—but we believe in your ability to explore and grow. Feel free to use Google, YouTube, or any learning resource. Learning how to learn is your biggest strength.

🛠️ Debug Yourself:

Mistakes? Perfect. That's how real learning happens. Try solving issues on your own—it'll build your confidence and sharpen your problem-solving skills for the future.

💸 No Paid Tools:

We value learning over luxury. If any task points to a paid tool, skip it. Don't spend a single rupee. Just search for free, open-source options—we promise, it's part of the real-world hustle.

📁 GitHub Submission:

For each task, start fresh. Create a new GitHub repo.

Upload everything you used—your code, dataset, screenshots (if any), and a short README.md. That README is your story—tell us what you built, why, and how.

📩 Submission:

When you're done and proud of what you've built, drop your GitHub repo link through the submission form. Let your work speak for you.