Day 1: Know the Battlefield

Goals:

Understand the big picture of your thesis.

Get all materials (code, report, slides) organized.

Tasks:

Ask teammates:

What is the exact problem we're solving? (e.g., defending against adversarial attacks on MNIST/CIFAR-10?)

What dataset is used? (MNIST? CIFAR-10? ImageNet?)

What attacks are we defending against? (FGSM? PGD?)

What is already implemented vs. what's left?

Get access to:

Codebase (GitHub repo? Google Drive?)

Report draft (if any)

Slides (if any)

Write a 1-page summary answering:

Problem Statement: What is Defense-GAN, and why is it useful?

Our Goal: What does your team's implementation aim to achieve?

Current Progress: What's working? What's not?

Day 2: GAN Foundations

Goals:

Understand how GANs work (generator vs. discriminator).

Learn the loss functions (min-max game).

Tasks:

Watch:

Ian Goodfellow's GAN Lecture (NIPS 2016)

Or DeepLearning.AI GAN Course (Coursera, free audit)

Read:

GAN Chapter from "Deep Learning with Python" (Chollet)

Key Questions:

How does the generator create fake data?

How does the discriminator detect fakes?

What is the minimax loss?

Output:

Draw a simple diagram of GAN training.

Write 1-2 paragraphs explaining GANs in your own words.

Day 3: Defense-GAN Theory

Goals:

Understand how Defense-GAN works (reconstruction + adversarial defense).

Tasks:

Read:

Defense-GAN Paper (Samangouei et al.)

Focus on Abstract, Section 3 (Methodology), Figures 1 & 2

Key Questions:

How does Defense-GAN reconstruct inputs?

Why does this help against adversarial attacks?

What is latent space optimization (Section 3.1)?

Output:

Write a 1-page summary explaining Defense-GAN in your own words.

Day 4: Adversarial Attacks Overview

Goals:

Learn why adversarial attacks are a problem.

Understand FGSM & PGD attacks (common attacks Defense-GAN defends against).

Tasks:

Read:

Explaining FGSM & PGD (Towards Data Science)

Adversarial Examples (OpenAI)

Key Questions:

What is an adversarial perturbation?

How does FGSM work? (Single-step attack)

How does PGD work? (Iterative attack)

Output:

Write a comparison table of FGSM vs. PGD.

Day 5: Code Day 1 – Architecture Walkthrough

Goals:

Understand code structure (where is generator/discriminator?).

Tasks:

Ask teammates:

Which file contains the GAN architecture?

Where is the training loop?

Where is the adversarial defense logic?

Draw a code map:

Example:

📂 Project

├── train.py (GAN training)

├── defense.py (Defense-GAN logic)

├── attacks/ (FGSM/PGD code)

└── eval.py (Testing defense)

Output:

A diagram of code structure.

Day 6: Code Day 2 – Train and Evaluate

Goals:

Run the code yourself and observe training.

Tasks:

Run training in Colab/local:

Try modifying batch size, learning rate.

Observe loss curves (does the GAN converge?).

Test Defense-GAN:

Generate an adversarial example (FGSM).

See if Defense-GAN detects/reconstructs it.

Output:

A short report on training behavior.

Day 7: Midpoint – Practice Explaining

Goals:

Verbalize your understanding.

Tasks:

Explain to a friend:

What is Defense-GAN?

How does it defend against attacks?

What's your team's contribution?

Output:

A presentation outline (your part).

Day 8: Deepen GAN + Read Related Work

Goals:

Learn limitations of Defense-GAN.

Tasks:

Read:

Defense-GAN Limitations (Section 5)

MagNet (Another Defense Method)

Output:

A list of limitations (e.g., slow reconstruction).

Day 9: Polish Presentation + Add Contribution

Tasks:

Improve slides:

Add a diagram of Defense-GAN workflow.

Write a clear summary of your team's work.

Day 10: Mock Defense

Tasks:

Practice Q&A:

"Why use GANs for defense?"

"How does reconstruction help?"

"What are the limitations?"

Days 11-14: Buffer & Final Polish

Rehearse presentation.

Fix last-minute bugs.

Get enough sleep!

Final Tips

☑ Use AI to explain code:

Ask ChatGPT: "Explain this Defense-GAN code snippet:" (paste code)

☑ Daily review: Spend 30 mins before bed summarizing what you learned.

☑ Stay focused: Stick to the plan—you got this!