Nontando Myoli

🔹 **Q1: How do AI-driven code generation tools (like GitHub Copilot) reduce development time? What are their limitations?**

How they save time:

- Suggest complete lines or blocks of code as you type.
- Help with boilerplate code (like loops, functions, and configs).
- Offer code examples instantly, saving time Googling.
- Speed up writing tests and documentation.

But they're not perfect:

- Sometimes generate incorrect or insecure code.
- Can't always understand the full logic of your project.
- Might produce code like existing open-source code (licensing risks).
- Developers may become too reliant and stop thinking critically.

🔹 **Q2: Compare supervised and unsupervised learning in automated bug detection**

	Supervised Learning 🧠✅	**Unsupervised Learning 🧠❓**
Data	Needs labelled examples (bug/no bug)	Works on unlabelled code
How it works	Learns patterns from known bugs	Detects anomalies in code behaviour
Good at	Spotting known issues reliably	Catching weird or new patterns
Weakness	Can't handle new/unknown bugs	Might give false alarms (false positives)
Example use	Classifying a known bug from history	Spotting unusual changes in new commits

---

🔹 **Q3: Why is bias mitigation important in AI-powered user experience personalization?**

- To keep recommendations **fair and inclusive** for all users.
- Without it, certain users might get **less relevant or unfair content**.
- Helps build **trust** with users and avoids hurting their experience.
- Prevents **legal or ethical issues** (especially in sensitive areas like job ads or credit suggestions).
- Protects your brand from backlash or accusations of discrimination.

---

2. Case Study: AI in DevOps – Automating Deployment Pipelines

How does AIOps make deployment more efficient?

AIOps = AI + DevOps
It uses machine learning and data to improve how we test, deploy, and monitor software.

Benefits of AIOps:

- **Predicts failures before they happen:**
  - It learns from past bugs and errors in your deployment pipeline.
  - If it sees something fishy (like a pattern linked to previous crashes), it warns the team early — saving time and headaches.
- **Makes pipelines smarter and self-healing:**
  - Automatically skips or fixes failing steps during deployment.
  - Adjusts which tests to run based on recent code changes.
  - Can even **auto-rollback** a broken release without manual input!