Part 1: Theoretical Analysis

1. Q1: AI Code Generation Tools

Prompt: Explain how GitHub Copilot reduces dev time. What are its limitations?

Sample Answer (Shortened):
GitHub Copilot reduces development time by suggesting code snippets in real time, reducing boilerplate coding and helping developers explore unfamiliar libraries. It accelerates prototyping and helps junior devs learn by example.

Limitations:

- ❖ May suggest incorrect or insecure code.
- ❖ Lacks full understanding of business logic.
- ❖ Relies on user input quality for context.

Q2: Supervised vs Unsupervised in Bug Detection

Supervised Learning:

- ❖ Uses labeled data (e.g., bug/no bug).
- ❖ Accurate if data is well-annotated.
- ❖ Good for predicting bug-prone files.

Unsupervised Learning:

- ❖ Uses unlabeled data (e.g., clustering unusual patterns).
- ❖ Good for anomaly detection and unknown bugs.

Less accurate but uncovers novel issues.

Q3: Bias in Personalization

AI personalization can reinforce existing biases if training data lacks diversity (e.g., user behavior skewed toward one group). This leads to unfair experiences. Bias mitigation ensures inclusivity and fairness.

Use tools like: IBM AI Fairness 360 to audit models and retrain on balanced data.

2. Case Study: AIOps in DevOps

Sample points to cover:

- AIOps monitors logs & metrics to detect deployment risks.
- Reduces downtime via anomaly prediction.
- **Example 1:** Auto rollback after failed deployment.
- **Example 2:** Intelligent load balancing during traffic spikes.