Part 1: Theoretical Analysis

Q1. Explain how AI-driven code generation tools (e.g., GitHub Copilot) reduce development time.

- Write and autocomplete code snippets, reducing repetitive tasks

-Suggest functions and syntax as you type

-Offer explanations for unfamiliar or complex code

What are their limitations?

- May generate incorrect or insecure code

-Might not adapt well to unique or unconventional project setups

-Risk of reducing deep understanding and coding independence

Q2: Compare supervised and unsupervised learning in the context of automated bug detection.

Supervised Learning

-Trains on labelled data (e.g., known bugs)

-Best for identifying familiar, well-documented errors

Unsupervised Learning

-Finds unusual patterns or outliers without labelled examples

-Useful for discovering new, unexpected bugs

Use **supervised** models for precision and **unsupervised** ones for exploration.

Q3: Why is bias mitigation critical when using AI for user experience personalization?

-Biased systems can unfairly favour or exclude individuals

-Can damage user experience, brand reputation, and trust

-May violate data ethics principles or legal frameworks

-Ensuring fairness boosts inclusivity, accountability, and reliability.

Case Study Analysis

How does AIOps improve software deployment efficiency? Provide two examples.

AIOps (Artificial Intelligence for IT Operations) streamlines deployment by automating detection and resolution of issues.

E.g. 1: Predicts performance degradation or outages ahead of time
E.g. 2: Automatically rolls back faulty releases or tunes system behaviour in real time