## **Part 1: Theoretical Analysis (30%)**

### **Q1: AI-Driven Code Generation Tools**

**Answer:** AI code generation tools like **GitHub Copilot** reduce development time by suggesting code snippets in real-time, automating boilerplate writing, and even generating entire functions from comments. Developers save time on syntax, API calls, and common algorithms, allowing them to focus more on logic and design.

**Limitations:**

* May generate incorrect or insecure code.
* Lacks deep contextual understanding.
* May replicate biased or low-quality training data.
* Requires constant human oversight and validation.

### **Q2: Supervised vs Unsupervised Learning in Bug Detection**

| **Aspect** | **Supervised Learning** | **Unsupervised Learning** |
| --- | --- | --- |
| Approach | Uses labeled bug data (e.g., bug/no-bug). | Detects anomalies or clusters without labels. |
| Use Case | Classify known bug types. | Discover unusual behavior (potential bugs). |
| Limitation | Needs large labeled datasets. | May flag false positives. |

**Example:**

* **Supervised**: Random Forest classifies crash logs.
* **Unsupervised**: K-means clusters abnormal system behavior.

### **Q3: Importance of Bias Mitigation in Personalization**

Bias in AI-driven personalization leads to:

* **Discrimination** (e.g., biased content recommendations).
* **Exclusion** of user groups.
* **Echo chambers**, reinforcing stereotypes.

**Bias mitigation** ensures fairness, diversity, and inclusivity. It prevents algorithms from favoring one demographic over another and supports ethical and user-centric software design.

### **Case Study: AIOps in DevOps**

**How AIOps improves deployment:**

1. **Anomaly detection**: AI identifies unusual behavior during deployment (e.g., error spikes).
2. **Automated rollback**: Machine learning models detect failures and trigger recovery scripts.

**Examples:**

* AI auto-scaling cloud instances based on traffic predictions.
* AI-powered root cause analysis reducing debugging time.