

Building Intelligent Software Solutions ■ ■

Objective: Demonstrate understanding of AI applications in software engineering through theoretical analysis, practical implementation, and ethical reflection.

Part 1: Theoretical Analysis (30%)

Q1: Explain how AI-driven code generation tools (e.g., GitHub Copilot) reduce development time. What are their limitations?

AI-driven code generation tools like GitHub Copilot reduce development time by suggesting context-aware completions based on large-scale models. They automate boilerplate coding and increase efficiency. Limitations include possible inaccuracies, biased or insecure code, overreliance, and limited understanding of project context.

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

Aspect	Supervised Learning	Unsupervised Learning
Definition	Uses labeled data to classify bugs.	Finds hidden patterns in unlabeled data.
Example Use	Predicting bug-inducing commits.	Clustering anomalies in logs.
Advantage	High accuracy when labeled data exists.	Useful when labels are unavailable.
Limitation	Needs large labeled datasets.	May flag false positives.

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

Bias mitigation ensures fairness and inclusivity. Without it, personalization may reinforce stereotypes or exclude users. Balanced datasets and fairness audits ensure equitable experiences for all demographics.

Case Study: AIOps in DevOps – Automating Deployment Pipelines

AIOps enhances deployment efficiency by automating anomaly detection and predictive scaling. Examples include predictive rollback before failures and dynamic resource allocation based on workload forecasts.

Part 2: Practical Implementation (60%)

Task 1: AI-Powered Code Completion

Copilot suggested code handles missing keys safely, while the manual version assumes consistent structure. Copilot’s version is slightly more robust, although both perform with $O(n \log n)$ complexity.

Task 2: Automated Testing with AI

Using Selenium and Testim.io, tests for login validation were automated. AI tools adapt to UI changes, increasing coverage and reducing maintenance compared to manual testing.

Task 3: Predictive Analytics for Resource Allocation

A Random Forest model achieved 0.96 accuracy and 0.95 F1-score on the Breast Cancer dataset. It predicts task priority effectively, showing AI’s utility in resource planning.

Part 3: Ethical Reflection (10%)

Bias in predictive models may affect fairness across teams. IBM AI Fairness 360 can detect and mitigate these biases through reweighing and adversarial debiasing, promoting ethical AI use.

Bonus Task: AI Documentation Generator (DocuMind)

DocuMind automatically generates documentation from code comments and commit messages using NLP. Integrated into GitHub Actions, it ensures up-to-date and accessible documentation, improving team productivity.