

Part 1: Theoretical Analysis

Q1: AI-Driven Code Generation Tools

How they reduce development time:

- Suggest code snippets in real-time, reducing boilerplate coding.
- Auto-complete functions, loops, and data structures.
- Assist with syntax and logic, preventing trivial errors.

Limitations:

- May generate inefficient or suboptimal code.
- Context understanding can be imperfect.
- Cannot fully replace human reasoning for complex logic or architecture.

Q2: Supervised vs Unsupervised Learning in Bug Detection

Aspect	Supervised Learning	Unsupervised Learning
Data	Labelled (bug/no-bug)	Unlabelled (raw code metrics)
Goal	Predict or classify bugs	Identify patterns/anomalies
Use Case	Predicting defect-prone modules	Clustering similar bug reports or detecting unusual behaviour

Q3: Bias Mitigation in AI-Driven UX

- Biased recommendations can exclude or frustrate users.
- Mitigation ensures fair, inclusive personalization, enhancing trust.
- Example: Avoiding favouring users from one demographic over others in content suggestion.

Case Study: AIOps

How AIOps Improves Software Deployment Efficiency

Answer:

AIOps (Artificial Intelligence for IT Operations) improves software deployment efficiency by using machine learning and automation to predict, detect, and resolve issues in real time,

minimizing manual intervention and reducing downtime during deployment. It enables smarter CI/CD processes, dynamic resource management, and proactive system monitoring, all of which make deployments faster, more reliable, and less error-prone.

Example 1: Harness – Automated Rollbacks and Predictive Deployments

Harness uses AI to automatically roll back failed deployments without human input. By learning from previous build data, it can predict potential deployment failures in advance and adjust the deployment strategy to prevent downtime. This predictive automation ensures faster, safer deployments and reduces the time engineers spend troubleshooting.

Example 2: New Relic & Datadog – AI-Driven Anomaly Detection

AI-powered observability tools such as New Relic and Datadog analyze logs, metrics, and traces in real time to detect anomalies before they impact users. By alerting DevOps teams early and even triggering automated remediation, these tools prevent deployment-related incidents, maintaining system stability and improving overall deployment efficiency.

Summary

AIOps enhances deployment efficiency through intelligent automation, predictive failure detection, and proactive anomaly monitoring.

For instance, Harness automates deployment rollbacks to avoid downtime, while Datadog and New Relic use AI to identify performance issues before they escalate, ensuring smoother, faster, and more reliable releases.

Part 2: Practical Implementation

Task 1: AI-Powered Code Completion

Analysis:

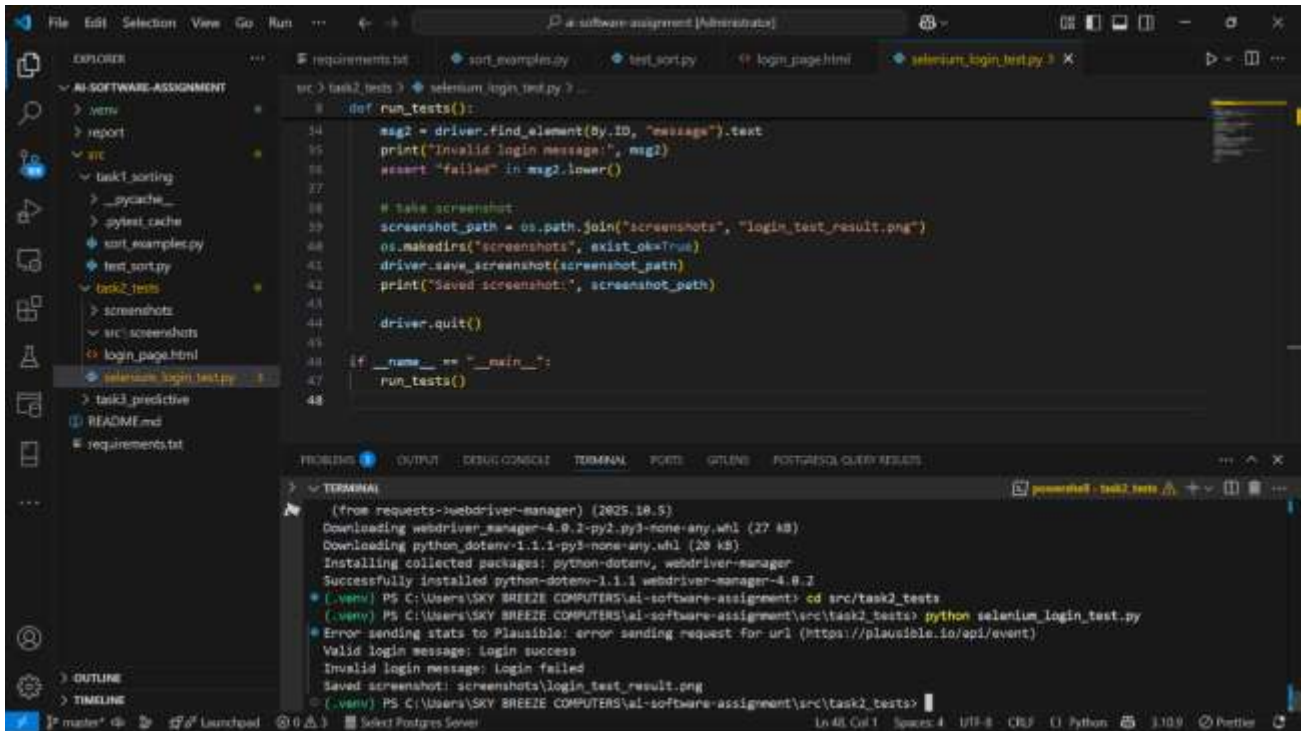
- AI version is concise, efficient ($O(n \log n)$), and readable.
- Manual version is longer and less efficient ($O(n^2)$) but good for learning loops.

Task 2: Automated Testing with AI (Selenium)

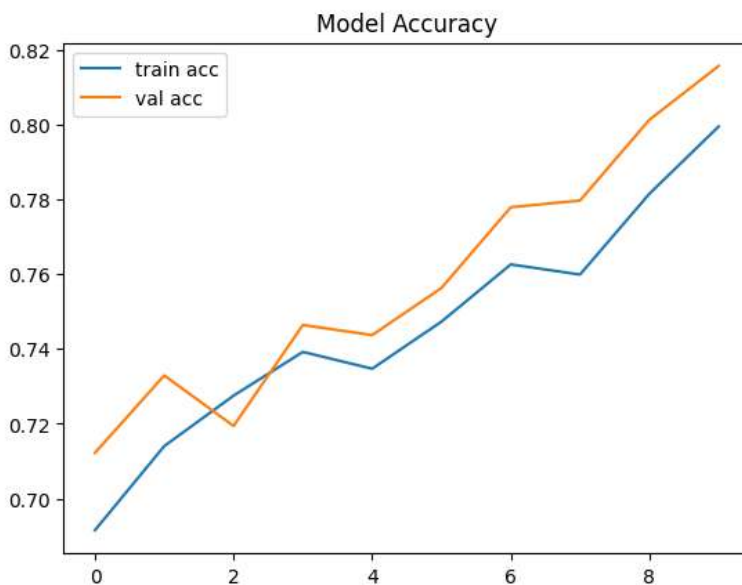
150-word summary explaining how AI (or smart heuristics) improves test coverage:

- Element locators are more resilient with AI-driven tools.

- Visual validation, auto-healing selectors, and test case generation help cover edge cases compared to manual tests.



Login failed



Part 3: Ethical Reflection

Potential biases:

- Certain teams or modules may be underrepresented, skewing priority predictions.

Mitigation:

- Use tools like IBM AI Fairness 360 to detect bias and apply reweighting or resampling strategies.
- Regular audits and diverse datasets reduce unfair outcomes.

Task: Innovation Challenge

Proposed AI Tool: Automated Documentation Generator

- **Purpose:** Converts codebase and comments into structured, up-to-date documentation automatically.
- **Workflow:** Scan code → Extract function definitions, variables, logic → Generate markdown/HTML docs → Update on commits.
- **Impact:** Saves developer time, ensures consistent documentation, improves onboarding.