a)	Present the problem situation in 5-7 sentences.	When a cybersecurity incident occurs such as a ransomware attack, data breach, or brute force intrusion organizations rely on incident response playbooks to guide their teams through detection, containment, eradication, and recovery. Currently, these playbooks exist as static Word documents, PDFs, or wiki pages that team members must read and follow manually under extreme time pressure. This manual approach leads to skipped steps, miscommunication, delayed escalations, and inconsistent responses. There is no way to test or validate a playbook before a real incident occurs. Existing automation tools SOAR platforms address this through drag-and-drop visual interfaces, but these are vendor-locked, expensive, and cannot be version-controlled, peer-reviewed, or tested like code. Security teams need a way to express their response procedures in a format that is both human-readable and machine-executable, bridging the gap between policy documentation and operational automation
b)	Formulate the problem – 1 statement.	Incident response playbooks are currently static, untestable documents that cannot be validated or executed automatically, leading to slow, inconsistent, and error-prone responses during cybersecurity incidents
TYPE OF QUESTION	**EXAMPLES OF QUESTIONS**	**SHORT ANSWER**
Who…?	1. Who has a problem? 2. Who is involved in the problem situation? 3. Who suffers from the problem? 4. Who should take part in solving the problem?	1. Security Operations Center (SOC) teams, incident responders, and CISOs who are responsible for handling cybersecurity incidents

		2. SOC analysts, incident response leads, forensics specialists, IT administrators, compliance officers, and management who approve and audit response procedures 3. The entire organization because delayed or inconsistent incident response leads to greater financial damage, data loss, regulatory penalties, and reputational harm 4. Security engineers, DSL/language designers, and domain experts in incident response who can define the correct playbook structures and validate the language design
What …? Which…?	5. What happened? 6. What is the problem? 7. What led you to identify the problem (reasons, causes, evidence, arguments, findings, survey results, etc.)? 8. What are the causes that affect those involved? 9. What are the needs of those targeted? 10. What are the resources (financial, human, logistical, time, etc.) necessary to solve the problem?	5. Incident response procedures remain trapped in static documents while every other part of the security pipeline like detection, monitoring, alerting has been automated. 6. There is no dedicated language that allows IR teams to write, validate, and execute incident response playbooks as code. 7. Analysis of existing IR workflows shows that playbooks from organizations like CISA and CERT Société Générale are distributed as PDFs and Word documents. Industry reports indicate that the average breach takes over 100 days to identify and remediate, partly due to manual, unstructured response processes

		8. Lack of automation tools accessible to non-developers, reliance on static documentation, absence of pre-execution validation, vendor lock-in of existing SOAR platforms, and high cost of commercial solutions 9. A simple, readable, domain-specific language that allows writing playbooks in familiar IR terminology, with static validation, automated execution, team configuration, and report generation without requiring programming expertise 10.A team of 4 developers, one semester of development time, Python as the host language, the Lark parser library, and access to publicly available playbook templates for reference and testing
Where…?	11. Where did the problem arise (location, area, stage of the process of the problem arising)?	11. In Security Operations Centers and incident response teams across organizations of all sizes, where the gap between written policy and operational execution remains unaddressed
When…?	12. When did the problem arise? 13. When was the problem identified? 14. When is it planned to be resolved?	The problem has existed since organizations began formalizing incident response procedures, but has become critical as cyberattacks have grown in frequency and sophistication over the past decade During the analysis phase of this project, when reviewing how real-world IR teams operate and comparing their static playbook documents against the automation

		capabilities available in other security domains By the end of the current semester, delivering a functional DSL prototype with parser, validator, interpreter, and example playbooks
Why…?	15. Why is there a need to solve the problem?	15. Because manual incident response is too slow and error-prone for modern cyber threats. Faster, validated, and automated response directly reduces breach impact, financial loss, and recovery time
How...?	16. How (in what way) can the problem be solved? 17. How will the elimination of the problem be determined?	16. By designing and implementing IncidentFlow a domain-specific language that allows IR teams to write playbooks as executable code with phases, triggers, conditionals, escalation chains, and automated report generation 17. By demonstrating that playbooks written in IncidentFlow can be statically validated for correctness, executed against simulated incidents, and produce accurate timeline reports all without requiring the user to know a general-purpose programming language
How much...?	18. How big is the problem? 19. How much is the quantity/quality/number of people etc. affected? 20. How long will it take to resolve the problem? 21. How many parties need to be involved in resolving it?	18. How big is the problem? It affects virtually every organization with a cybersecurity function. Even large enterprises with SOAR tools struggle with playbook maintenance and validation. 19. Thousands of SOC teams globally. According to industry

		data, the cybersecurity workforce gap exceeds 3.5 million people 20.The prototype will be developed over one semester 21.A team of 4 members covering language design, parser implementation, execution engine, validation, testing, and documentation
Other questions you found answered (optional)	22. 23.	-
	How do you see solving this problem (in terms of deliverable/product/result)?	a working DSL with a parser, static validator, simulated execution engine, and report generator, accompanied by three example playbooks and a comprehensive project report documenting the design and implementation
	Conclusions, findings, reflections...	Current incident response practices rely heavily on human interpretation of static documents under high-stress conditions. While SOAR platforms partially address this, they are expensive, vendor-locked, and GUI-based making them impossible to version-control or peer-review

PBL PROBLEM MAP