

# SIMPLE BUSINESS RISK REGISTER TEMPLATE

PROBABILITY	5	5	10	15	20	25
	4	4	8	12	16	20
	3	3	6	9	12	15
	2	2	4	6	8	10
	1	1	2	3	4	5
		1	2	3	4	5
	IMPACT					

RISK DESCRIPTION	IMPACT DESCRIPTION	IMPACT LEVEL	PROBABILITY LEVEL	PRIORITY LEVEL	MITIGATION NOTES	OWNER
Give a brief summary of the risk.	What will happen if the risk is not mitigated or eliminated?	Rate 1 (LOW) to 5 (HIGH)	Rate 1 (LOW) to 5 (HIGH)	(IMPACT X PROBABILITY) Address highest first.	What can be done to lower or eliminate the impact or probability?	Who's responsible?
Navbar library wasn't compatible with react-router-dom.	Will not render to the correct paths to display all corresponding info.	5	5	25	Stop use of multilevel navbar & implement a manual navbar.	Front-End Team
Client might resort to mobile-first design.	Aspects of screen may not be visible to user.	5	2	10	Regular meetings with client to demonstrate the prototype & get feedback.	Product Owner
Absence due to sickness/ other commitments within the team.	Slows down development of final app, especially with a deadline.	3	4	12	Keep communication high throughout the team with all updates.	Full-Stack Team
Uploading secret API key to public repositories.	Leads to security risk & unauthorised access to sensitive data & services.	5	3	15	Restrict API key usage & add access control. Use a secure config system.	Front-End Team/ DevOps/Security Team

KEY

LEVEL

1

2

3

4

Data Security Breach	Access to & leaking of confidential data is available to anyone, safeguarding issues.	5	2	10	Understand & comply with data protection laws/regulations applicable to project. Conduct regular audits.	Legal/Compliance Team	5
Insufficient User Training	Some users may struggle to use the employee scheduling system effectively, leading to errors.	3	4	12	Provide comprehensive user training sessions & documentation before system launch.	Training Manager	
Inadequate testing of all functions of full software system.	Insufficient test coverage can result in undetected bugs in the system during production use.	4	2	8	Implement comprehensive testing strategies of each functionality before implementing new features.	Full-Stack Team	
Lack of scalability	HR system scheduling system may be unable to handle increased user loads, leading to performance degradation.	3	2	6	Design system with scalability in mind, making functionality reusable & no hard coding. Increase testing & optimise infrastructure.	Full-Stack Team/System Architect	
Inadequate Backlog Management	Poorly managed backlog items may lead to unclear priorities & hinder efficient development of critical features.	2	2	4	Implement agile backlog management practices & prioritise items based on functionality & business value.	Scrum Master	

--	--	--	--	--	--	--	--