T	Task 1(A) - The Proposal					
		1-3	4-6	7-9		
	Decompose Problem	• Identifies <b>some</b> problems to be	Identifies most problems to be solved	Proposal:		
		solved	Decomposes most of the problems to be solved	Fully Identifies problems to be solved		
		• Decomposes <b>some</b> of the	Dunancial adjustice officiatively.	Effectively decomposes the problems to be		
		problems to be solved	Proposed solution effectively:  • Meets most client/user needs	solved		
		<ul> <li>Meets some client/user needs</li> </ul>	Mitigates most potential risks	Proposed solution effectively:		
	coi	Mitigates <b>some</b> potential risks	addresses most relevant regulatory guidelines & legal	Meets full needs of client/users		
	De	<ul> <li>addresses some relevant</li> </ul>	requirements	Mitigates <b>all</b> potential risks		
		regulatory guidelines & legal		addresses all relevant regulatory guidelines &		
		requirements		legal requirements		
	Wider Issues	<b>Limited</b> lines of reasoning that	Good lines of reasoning that mostly justify how:	Comprehensive lines of reasoning that fully justify		
		partially justify how:	Solution meets the needs of the client and users	how:		
		<ul> <li>Solution meets the needs of the client and users</li> </ul>	Potential risks will be mitigated     Solution will address relevant regulators guidelines.	Solution meets the needs of the client and users     Determined ricks will be mitigated.		
,		Potential risks will be mitigated	<ul> <li>Solution will address relevant regulatory guidelines and legal requirements</li> </ul>	<ul><li>Potential risks will be mitigated</li><li>Solution will address relevant regulatory</li></ul>		
	<u>s</u> ≤	Solution will address relevant	and legal requirements	guidelines and legal requirements		
		regulatory guidelines and legal		gardennes and regar requirements		
		requirements				
		Basic definitions of:	Good definitions of:	Comprehensive & perceptive definitions of:		
	ess	<ul> <li>functional and non-functional</li> </ul>	functional and non-functional requirements	functional and non-functional requirements		
	susiness Context	requirements	key performance indicators	key performance indicators		
	Business	<ul> <li>key performance indicators</li> </ul>	user acceptance criteria	user acceptance criteria		
		<ul> <li>user acceptance criteria</li> </ul>				

T	ask 1(B)	) - The Design					
		1-3	4-6	7-9			
	S	Interface is adequate as a result of reasonably	Interface is <b>good</b> as a result of <b>effective</b> use of:	Interface is <b>excellent</b> as a result of <b>sophisticated</b> and			
	ace	effective use of:	layout and white space	highly effective use of:			
	ver	layout and white space	visual hierarchies	layout and white space			
	ffectiveness of Interface	visual hierarchies	common conventions	visual hierarchies			
	Effectiveness of Interface	• common conventions		• common conventions			
	a	Basic decomposition of problems that superficially	Good decomposition of problems that superficially	Highly effective decomposition of the problems that			
	soc em	cover:	cover:	superficially cover:			
	ecompos problem	• inputs	• inputs	• inputs			
	Decompose problem	• processes	• processes	• processes			
	۵	• outputs	• outputs	• outputs			
		Algorithms produce some correct outcomes as a	Algorithms produce mostly correct outcomes as a	Algorithms produce <b>consistently</b> correct outcomes as			
	<u>જ</u>	result of:	result of:	a result of:			
	cing ons	• some precise logic	mostly precise logic	• precise logic			
	nin nti	• some appropriate structure and sequence which is	appropriate structure and sequence but may lack	efficient structure and sequence			
	Logical thinking conventions	likely to be <b>inefficient</b>	efficiency				
	gica cor						
	Log	<b>Some</b> effective use of accepted conventions	Mostly effective use of accepted conventions though	Effective and consistent use of accepted conventions			
		although inconsistencies still exist.	some minor inconsistencies may still exist.				
	uirements	Somewhat appropriate, including (as required):	Mostly appropriate, including (as required):	Fully appropriate, including (as required):			
		• variables	• variables	• variables			
		data structures	data structures	data structures			
		data types	data types	data types			
		Naming conventions <b>mostly</b> appropriate but	Naming conventions appropriate and mostly	Thoroughly appropriate and consistent naming			
	a F	inconsistent.	consistent.	conventions are used throughout.			
	Dat	Effective error handling identified for <b>some</b>	Effective error handling identified for most	<b>Thoroughly</b> effective error handling identified for			
		inputs/processes.	inputs/processes.	inputs/processes.			
	>	Basic understanding of:	Good understanding of:	Thorough & detailed understanding of:			
	eg	how components interrelate	how components interrelate	how components interrelate			
	rat	• the order in which components should be tested	• the order in which components should be tested	the order in which components should be tested			
	t SI	• the types of test that are required.	• the types of test that are required.	the types of test that are required.			
	Test Strategy						

	2
	Comminication
-	-
of	7
	- ;'
~	.=
ality	2
=	=
<u>0</u>	-
Q	2
$\boldsymbol{\alpha}$	_
_	2
	C
	C
	_

**Some** effective communication as a result of:

- some appropriate techniques, methods & formats
- **some** technical language, **appropriate** for intended audience

**Mostly** effective communication as a result of:

- mostly appropriate techniques, methods & formats
- technical language, **mostly** appropriate for intended audience

Communication **consistently** effective as a result of:

- **consistently** appropriate techniques, methods & formats
- technical language, **consistently** appropriate for intended audience

Task 2 –	Task 2 – Developing the Solution (The Solution)				
	1-2	3-4	3-5?	6-8	
	Implemented in a single language with	Implemented with some functionality	Functional code in at least two	Functional code in at least two	
<u>₹</u>	some functionality. Code lacks	in at least two languages. Code lacks	languages. Mostly efficient with	languages. Code is consistently	
Functionality	efficiency with some major errors.	efficiency with some major errors.	Some minor errors.	efficient.	
ţi					
ın C	Some precise logic and structures	Sufficient precise logic and structures	Uses mostly precise logic and	Precise logic and structures throughout	
교	which result in <b>some</b> correct	which result in <b>adequate</b> correct	structures which result in mostly	which result in <b>consistently</b> correct	
	outcomes.	outcomes.	correct outcomes.	outcomes.	
_	Maintainable by a third party, but with	Maintainable by a third party, but with	Maintainable by a third party, with	Easily maintainable by a third party with	
io	significant difficulties due to:	some difficulties due to:	only a few minor difficulties due to:	consistently appropriate:	
Code Organisation	• inconsistent naming conventions.	<ul> <li>somewhat appropriate naming</li> </ul>	<ul> <li>mostly appropriate naming</li> </ul>	naming conventions.	
Can	limited logical organisation	conventions.	conventions.	logical organisation	
Org	• limited informative commenting.	<ul> <li>some logical organisation</li> </ul>	<ul> <li>mostly logical organisation</li> </ul>	informative commenting.	
		<ul> <li>some informative commenting.</li> </ul>	<ul> <li>mostly informative commenting.</li> </ul>		
	Basic user experience through limited	Adequate user experience through	Good user experience through mostly	Excellent user experience through	
9	effective use of:	somewhat effective use of:	effective use of:	consistently effective use of:	
experience	input handling	<ul><li>input handling</li></ul>	• input handling	input handling	
per	user guidance and error messages	<ul> <li>user guidance and error messages</li> </ul>	<ul> <li>user guidance and error messages</li> </ul>	user guidance and error messages	
	• outputs	• outputs	• outputs	• outputs	
User	Partially robust and effectively handles	Adequately robust and effectively	Largely robust and effectively handles	Fully robust and effectively handles	
Š	some common errors.	handles sufficient common and	most common and unexpected errors	common and unexpected errors	
		unexpected errors			
	<b>Some</b> effective application of	Mostly effective application of	Consistent and effective application of		
SL	standards and guidelines in relation to:	standards and guidelines in relation to:	standards and guidelines in relation to:		
tior	accessibility	<ul><li>accessibility</li></ul>	<ul><li>accessibility</li></ul>		
ulat dar	compatibility	<ul><li>compatibility</li></ul>	<ul><li>compatibility</li></ul>		
egr	legal and ethical considerations	<ul> <li>legal and ethical considerations</li> </ul>	<ul> <li>legal and ethical considerations</li> </ul>		
al r	<b>Some</b> effective procedures and	Mostly effective procedures and	Thoroughly effective procedures and		
Legal regulations & Standards	security controls to ensure	security controls to ensure	security controls to ensure		
	confidentiality, integrity & availability.	confidentiality, integrity and	confidentiality, integrity and		
		availability.	availability.		

Task	Task 2 – Developing the Solution (Testing & Documentation)					
	1-2	3-4	5-6			
	<b>Basic</b> understanding of how to effectively test inputs,	Good understanding of how to effectively test	Thorough and detailed understanding of how to			
<u> </u>	calculations, validation and processes using test data	inputs, calculations, validation and processes using	effectively test inputs, calculations, validation and			
ig ?	which makes <b>limited use</b> of:	test data which includes:	processes using test data which includes:			
ita	• normal data	• normal data	normal data			
Suitability	• erroneous data	• erroneous data	erroneous data			
	extreme data	extreme data	extreme data			
	Basic understanding of how errors/problems were	Good understanding of how errors/problems were	Comprehensive understanding of how			
<b>b</b> 0	identified and how they were rectified for:	identified and how they were rectified for:	errors/problems were identified and how they were			
ti	• inputs	• inputs	rectified for:			
Testing	• calculations	• calculations	• inputs			
	validation and processes	<ul><li>validation and processes</li></ul>	calculations			
atj.			validation and processes			
Iterative	Testing shows evidence of a <b>basic</b> iterative	Testing shows evidence of a <b>good</b> iterative				
=	development process.	development process.	Testing shows evidence of an <b>effective</b> iterative			
			development process.			
a ,	Basic iterative development process, including:	Adequate iterative development process, including:	Effective iterative development process, including:			
ţį	Limited/superficial records of changes made	<ul> <li>Adequate recording of changes made</li> </ul>	Thorough & detailed recording of changes made			
Iterative	Superficial/vague reasons for changes made	<ul> <li>Supported reasons for some changes made</li> </ul>	Convincing & perceptive reasons for changes made			
= °	Some effective use of versioning	Mostly effective use of versioning	Mostly effective use of versioning			

Task 3 (A) – Gathering feedback to inform future developments				
	1-3	4-6	7-8	9-12
Effectiveness of materials	Materials allow for <b>limited</b> quality feedback for different aspects of the prototype.	Materials allow for <b>adequate</b> quality feedback for different aspects of the prototype.	Materials allow for <b>good</b> quality feedback for different aspects of the prototype.	Materials allow for <b>high</b> quality feedback for different aspects of the prototype.
Use of tools	Resulted in feedback that provides some opportunity for evidence-informed further iteration.	Resulted in feedback that <b>mostly</b> provides the opportunity for evidence-informed further iteration.	Resulted in feedback that <b>consistently</b> provides the opportunity for evidence-informed further iteration.	
Effectiveness of communication	Sometimes effective for both technical and non-technical audiences. Limited use of appropriate techniques, methods and formats.	Mostly effective for both technical and non-technical audiences. Use of mostly appropriate techniques, methods and formats.	Effective for both technical and non-technical audiences. Consistent use of appropriate techniques, methods and formats.	
#	Limited technical language appropriate for intended audience.	Technical language <b>mostly</b> appropriate for intended audience.	Technical language <b>consistently</b> appropriate for intended audience.	

Task 3 (B) – Evaluating feedback to inform future developments				
	1-2	3-4	5-6	
	<b>Limited</b> review of the content selected. including <b>superficial</b> consideration of:	<b>Good</b> review of the content selected. including <b>good</b> consideration of:	<b>Comprehensive</b> review of the content selected. including <b>thorough</b> consideration of:	
of int	• appropriateness	• appropriateness	• appropriateness	
sss	source validity/reliability	source validity/reliability	<ul><li>source validity/reliability</li></ul>	
Effectiveness of assets & content	legal and ethical implications	legal and ethical implications	legal and ethical implications	
fect	Sometimes supported by superficial consideration,	<b>Mostly</b> supported by <b>good</b> consideration, comparison	Well supported by effective consideration,	
ass	comparison and corroboration across multiple	and corroboration across multiple sources.	comparison and corroboration across multiple	
	sourceschecked? Verified? Delete and just use		sources.	
	'comparison?'			
	1-3	4-6	7-8	
	1-2	4-0	7-0	
	Basic/superficial evaluation of how well prototype meets:	Good evaluation of how well prototype meets:	Thorough & detailed evaluation of how well prototype	
	-	1 7		
	Basic/superficial evaluation of how well prototype meets:	Good evaluation of how well prototype meets:	Thorough & detailed evaluation of how well prototype	
es	Basic/superficial evaluation of how well prototype meets: • functional and non-functional requirements	Good evaluation of how well prototype meets:  • functional and non-functional requirements	Thorough & detailed evaluation of how well prototype meets:	
omes	Basic/superficial evaluation of how well prototype meets:     functional and non-functional requirements     key performance indicators (KPIs)     user acceptance criteria	Good evaluation of how well prototype meets:  • functional and non-functional requirements  • key performance indicators (KPIs)	Thorough & detailed evaluation of how well prototype meets:  • functional and non-functional requirements	
itcomes	Basic/superficial evaluation of how well prototype meets:  functional and non-functional requirements  key performance indicators (KPIs)	Good evaluation of how well prototype meets:  • functional and non-functional requirements  • key performance indicators (KPIs)	Thorough & detailed evaluation of how well prototype meets:  • functional and non-functional requirements • key performance indicators (KPIs)	
outcomes	Basic/superficial evaluation of how well prototype meets:	Good evaluation of how well prototype meets:  • functional and non-functional requirements  • key performance indicators (KPIs)  • user acceptance criteria	Thorough & detailed evaluation of how well prototype meets:  • functional and non-functional requirements • key performance indicators (KPIs)	
ect outcomes	Basic/superficial evaluation of how well prototype meets:	Good evaluation of how well prototype meets:  • functional and non-functional requirements  • key performance indicators (KPIs)  • user acceptance criteria	Thorough & detailed evaluation of how well prototype meets:  • functional and non-functional requirements • key performance indicators (KPIs) • user acceptance criteria	
roject outcomes	Basic/superficial evaluation of how well prototype meets:     functional and non-functional requirements     key performance indicators (KPIs)     user acceptance criteria  Basic/simplistic reasons for future iteration.  Supported by limited relevant:     selection of examples	Good evaluation of how well prototype meets:  • functional and non-functional requirements  • key performance indicators (KPIs)  • user acceptance criteria  Good reasons for future iteration.  Supported by mostly relevant:  • selection of examples	Thorough & detailed evaluation of how well prototype meets:  • functional and non-functional requirements • key performance indicators (KPIs) • user acceptance criteria  Convincing & perceptive reasons for future iteration.  Supported by entirely relevant & perceptive:	
Project outcomes	Basic/superficial evaluation of how well prototype meets:	Good evaluation of how well prototype meets:  • functional and non-functional requirements  • key performance indicators (KPIs)  • user acceptance criteria  Good reasons for future iteration.  Supported by mostly relevant:	Thorough & detailed evaluation of how well prototype meets:  • functional and non-functional requirements • key performance indicators (KPIs) • user acceptance criteria  Convincing & perceptive reasons for future iteration.  Supported by entirely relevant & perceptive: • selection of examples	
Project outcomes	Basic/superficial evaluation of how well prototype meets:     functional and non-functional requirements     key performance indicators (KPIs)     user acceptance criteria  Basic/simplistic reasons for future iteration.  Supported by limited relevant:     selection of examples	Good evaluation of how well prototype meets:  • functional and non-functional requirements  • key performance indicators (KPIs)  • user acceptance criteria  Good reasons for future iteration.  Supported by mostly relevant:  • selection of examples	Thorough & detailed evaluation of how well prototype meets:  • functional and non-functional requirements • key performance indicators (KPIs) • user acceptance criteria  Convincing & perceptive reasons for future iteration.  Supported by entirely relevant & perceptive:	
Project outcomes	Basic/superficial evaluation of how well prototype meets:     functional and non-functional requirements     key performance indicators (KPIs)     user acceptance criteria  Basic/simplistic reasons for future iteration.  Supported by limited relevant:     selection of examples	Good evaluation of how well prototype meets:  • functional and non-functional requirements  • key performance indicators (KPIs)  • user acceptance criteria  Good reasons for future iteration.  Supported by mostly relevant:  • selection of examples	Thorough & detailed evaluation of how well prototype meets:  • functional and non-functional requirements • key performance indicators (KPIs) • user acceptance criteria  Convincing & perceptive reasons for future iteration.  Supported by entirely relevant & perceptive: • selection of examples	