Mimi Masic

NEA Survey Response

The student

Name	Mimi Masic
School Email	sy17masica@sydenham.lewisham.sch.uk
Programming Level	4 / 10

Student's project

Description	A flashcard revision website which will sort the flashcards based on how well the client knows them, sort of quizlet-esque (hopefully...)
List of languages	Python, HTML, CSS, Java
List of technologies	Django
Experience using languages/technologies	Python - I have been using since year 7 but this doesn't mean I'm good at it... HTML - I am pretty decent at it, used it in year 8 and liked it and now using it for my project - feel pretty okay with it (and with some help from W3 schools) CSS - I can kind of use this - this project is the first time I have used it Java - ... Django - my first time using it but I am slowly getting the hang of it
Client	
Client's identity	My friend, Karina
Client fictional?	No

Student's Progress

Current section	Design
List of completed	Analysis

sections	
Current page count	Analysis - 6 Design - 2
Progress by section	
Analysis	75% < x < 100%
Design	25% < x ≤ 50%
Technical Implementation	0 < x ≤ 25%
Testing	25% < x ≤ 50%
Evaluation	Not started (0%)

Other

Implementation concerns	I think I am worried about implementing the actual flashcard learning feature
Anything else? (Misc)	I am not the most technical person when it comes to practical and tend to ask some rather stupid questions (I was presented a duck by mr Simpson...) but once I understand something I am able to do it very well

Louis' Comments

General Comments	Mimi has made decent progress, although her page count would ideally be a bit higher. Like a few of the other students, Mimi seems to have moved on to implementing her project before fully completing her design. Given the potential scope of her project, I'm slightly concerned this could lead to her over-scoping and not producing a working submittable project in time for the submission deadline. When she refers to Java in the list of languages, I presume she is referring to JavaScript. I can't believe Simpson has been giving ducks to other students. He should know that *a-duck-tery* is a sin. ***<hysterical laughter followed by rapturous applause>***

Next steps	To ensure that her documentation includes all of the necessary content, she could create a checklist using the exemplars. Given the scope of the project, ensuring that it is designed in such a way that it can be implemented in a modular manner is probably advisable. Ensure that she knows that Java and JavaScript are two different things and JS cannot be referred to as simply 'Java'. (Doing so could lead to a fiar amount of confusion when she tried to look up documentation.)
Complexity	If the project is completed, then it is likely to be in the top complexity band

See the next page for detailed complexity band information.

			Mimi Masic
BOTTOM MARK BAND	**Algorithms**	Simple mathematical calculations	Not Used
		Linear search	Not Sure
	Databases	Non-SQL table access	Not Sure
		Simple data structures	Not Sure
MIDDLE MARK BAND	**Algorithms**	Simple scientific/mathematical /robotics/control/business model	Not Used
		Bubble Sort	Not Sure
		Binary search	Not Sure
		Simple user defined algorithms	Not Used
	Databases	Single table database	Not Sure
		Simple data model in database	Must Have
		Writing and reading from files	Not Sure
	File Access	Text files	Not Sure
		File(s) organised for sequential access	Not Sure
	Web *Stuff*	Calling Web service APIs	Not Sure
		Simple client-server model	Not Sure
	Data Structures	Multi-dimensional arrays	Not Sure
		Dictionaries	Could Have
		Records	Could Have
		Simple OOP model	Could Have
TOP MARK BAND	**Algorithms**	Complex scientific/mathematical/robotics/control/business model	Not Used
		Hashing	Must Have
		Merge sort	Not Sure
		Advanced matrix operations	Not Sure
		Recursive algorithms	Not Sure
		Graph/Tree Traversal	Not Sure
		Complex user defined algorithms	Not Sure
	Databases	Complex data model in database	Must Have
	File Access	Files(s) organised for direct access	Not Sure
	Web *Stuff*	Server-side scripting using request and response objects	Must Have
		Complex client-server model	Not Sure
	Data Structures	Hash tables	Could Have
		Lists	Could Have
		Stacks	Could Have
		Queues	Could Have
		Graphs	Could Have
		Trees	Could Have
		Complex OOP model	Could Have
		Linked lists	Could Have