Jamie Whitworth

NEA Survey Response

The student

Name	Jamie Whitworth
School Email	whitj537.209@student.foresthillschool
Programming Level	6 / 10

Student's project

Description	An interactive timetable implemented into a revision app. you can make notes and flashcards which are stored under specific topics and set yourself them at specific days and times for a planned out structure. it will track your recent attempts and show your improvement/decline.
List of languages	python
List of technologies	python
Experience using languages/technologies	python - have been using since yr7
Client	
Client's identity	Ibrahim Kamara(a student struggling with time management/procrastination)
Client fictional?	No

Student's Progress

Current section	Design
List of completed sections	Analysis
Current page count	analysis - 10

Progress by section	
Analysis	25% < x ≤ 50%
Design	Not started (0%)
Technical Implementation	Not started (0%)
Testing	Not started (0%)
Evaluation	Not started (0%)

Other

Implementation concerns	
Anything else? (Misc)	

Louis' Comments

General Comments	Jamie's analysis page count is fairly good, however, he suggests he still has the majority of his analysis to complete. This, along with the fact he hasn't started to work on any of the other sections means his progress is concerning. I am concerned that Jamie may not have fully grasped the complexity of the project, as it involves scheduling. It's also worth noting that appears to be building a spaced repetition algorithm like the one used by Anki. He'll need to explain why his client can't just use Anki. What makes his app better?
Next steps	Encourage Jamie to move on to the design section as soon as possible and ensure that he is aware of how complex the implementation of certain bits of his project may be. Hopefully, he will get a sense of this complexity while completing his design section. It would be advisable that he compartmentalises his project, to ensure that he can produce a minimum viable product in time for the submission deadline.
Complexity	If complete, this project is likely to reach the top complexity band.

See the next page for detailed complexity band information.

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