

Neo

NEA Survey Response

The student

Name	Neo
School Email	weign004.209@student.foresthillschool.co.uk
Programming Level	6 / 10

Student's project

Description	Website for teachers and students to help revision. Make and use revision cards/quizzes and interact with others on the forum. Teachers can make classes and add students to them where they set them work and can view their progress and stats.
List of languages	Python maybe some javascript (html and css?)
List of technologies	Django
Experience using languages/technologies	python - on and off for 5 years but at the level of someone who has 1 year probably Django - 4 months but understand the basics and learn quick from gpt html - can get by css - can get by js - really bad
Client	
Client's identity	Martin
Client fictional?	Yes

Student's Progress

Current section	Design
List of completed sections	Analysis

Current page count	analysis 12
Progress by section	
Analysis	Completed (100%)
Design	Not started (0%)
Technical Implementation	$25\% < x \leq 50\%$
Testing	Not started (0%)
Evaluation	Not started (0%)

Other

Implementation concerns	the "cards/quizzes" app js deploying
Anything else? (Misc)	https://github.com/NeoWeight/revision_site (i am new to github so dont know if i am uploading anything that i shouldnt like private keys and stuff)

Louis' Comments

General Comments	<p>While the lack of any progress in the design section is not ideal, I'm fairly certain Neo has more content in his analysis section than anyone else in the class.</p> <p>I'm slightly concerned that Neo may be at risk of over-scoping and not ending up with a finished project. It might be a good idea if he chooses either the cards or quizzes and focuses on that. Especially given the fact he highlights these two functionalities as things he's worried about implementing.</p> <p>Neo's project can be found on GitHub.</p>
Next steps	<p>Neo mentions that he doesn't know much about keeping secrets safe while using GitHub, so it might be worth explaining how to use a config.py file and .gitignore.</p> <p>I'd recommend that Neo creates a high-level design for his project as soon as possible, so he's clear on what needs to be implemented. Given the time constraints, it might be a good idea for him to create some of his design stuff while implementing the project (e.g. if he is creating a new page, he</p>

	<p>can first create a labelled prototype using draw.io).</p> <p>As part of his design work, Neo should compartmentalise his project so that he can prioritise certain features and ensure he finishes the implementation of these before the submission deadline.</p> <p>Given that Neo identifies “deploying” as an area of concern, someone needs to tell him the project doesn't need to be deployed (for the NEA).</p>
Complexity	If completed, this project is likely to be in the top complexity band.

See the next page for detailed complexity band information.

			Neo
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	Should Have
		Bubble Sort	Not Sure
		Binary search	Not Sure
		Simple user defined algorithms	Must Have
	Databases	Single table database	Must Have
		Simple data model in database	Must Have
		Writing and reading from files	Should Have
	File Access	Text files	Should Have
		File(s) organised for sequential access	Not Sure
	Web Stuff	Calling Web service APIs	Could Have
		Simple client-server model	Should Have
	Data Structures	Multi-dimensional arrays	Could Have
		Dictionaries	Should Have
		Records	Not Sure
		Simple OOP model	Should Have
TOP MARK BAND	Algorithms	Complex scientific/mathematical/robotics/control/business model	Should Have
		Hashing	Could Have
		Merge sort	Not Sure
		Advanced matrix operations	Not Sure
		Recursive algorithms	Must Have
		Graph/Tree Traversal	Not Sure
		Complex user defined algorithms	Could Have
	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	Not Sure
		Complex client-server model	Could Have
	Data Structures	Hash tables	Not Sure
		Lists	Must Have
		Stacks	Should Have
		Queues	Should Have
		Graphs	Should Have
		Trees	Should Have
		Complex OOP model	Not Sure
		Linked lists	Could Have