Survey Builder Documentation

# Overview

This document aims to highlight the spec and progress of Survey Builder.

# Spec

* Select question modules and rearrange to build a survey.
* Create custom lists to input into questions (like brand list).
* Customisation wording of question.
* Verification of user changes to question.
* Delete/move questions within a module.
* Add questions to module from decipher.
* Timing tool for survey.

# Technical Notes:

Have created a QuestionCluster class which should house a cluster of questions that are meant to sit on the same page and have shared ‘execs’ etc.

When converting back to XML, should look in a QuestionCluster and if there are no Questions (because they have all been deleted) then it should just ignore the QuestionCluster.

Need to create a way to create these QuestionClusters.

Also need to create the parent link between terms, execs etc. to a Question or QuestionClustser.

To XML conversion once user modifications have been made should ideally be done via modules then via questions so the reordering of questions and removal of questions are accounted for. There will be some elements that are not tied to specific questions so these should be identifiable and still run.

Essentially there should be a clear tree of objects.

HTML -> QuestionCluster -> HTML -> Question -> …

Where terms, suspends etc. are nested within a QuestionCluster/Question

Thought: Can we create ID variables in the XML so we can link up things to each other?

# Frontend

* Make an design outline in Figma
* Figma creates a React frontend ‘.tsx’.
* It looks great but requires you to convert Python backend into an API
* The API would need to be hosted online and requests from the frontend would be made to it
* You can create a Python API using **Flask** or **FastAPI**.
* <https://www.pythonanywhere.com/> allows hosting a Flask API (potential ISO nightmare).
* AWS also allows for hosting

George To Do:

* Link up suspends directly to a question so they only exist within a question
  + Create parser class so we can keep track of historic parsed objects and manipulate them
* Add ability to move questions within a module