|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Module 5 PMA  Digital Development with Python | | | | | |
|  |  |  |  |  |
| Lecturers  Michael Mortenson  Jordan Bruno | | | Student ID: u2184618  Student name: Harold Celis  Student e-mail: [harold.celis-navarrete@warwick.ac.uk](mailto:harold.celis-navarrete@warwick.ac.uk)  24th march, 2022. | |

Abstract

Contents

[Abstract 2](#_Toc98665435)

[Introduction 5](#_Toc98665436)

[Application goals 6](#_Toc98665437)

[Application front-end 7](#_Toc98665438)

[Application back-end 8](#_Toc98665439)

[Conclusion 9](#_Toc98665440)

[References 10](#_Toc98665441)

Table of pictures

Table of charts

Table of tables

Introduction

The purpose of this written work is to document a webapp proposal that help students plan their dissertation project during their masters studies in University of Warwick, specially in the timeframe where students have to find a research topic and tutor. This phase accounts 50% of the entire process but impacts 100% of it. This means, the webapp does not focus on time and scheduling. It focus on motivation and organization of ideas, that if done well, will affect positively the dissertation schedule.

The scope of this written paper is a basic-level explanation of the proposed code structure, languages and functionality, which also show a basic layout for final users. That means that there is no feasibility, marketing, user experience or any other studies that are important for technology developments (Beck, 2000).

During the process of dissertation planning, students face many stages that when not completed at time, might result in other problems such as lack of interest in chosen subject, miss deadlines, accumulation of work, among other. For those reasons, I decided that a planning webapp is not completed without taking into account the level of enjoyment while planning, as well as the enjoyment in the chosen dissertation topic.

This written work is composed by three sections, application goals, application front-end and application back-end. First section, explain current problem with students and their planning process and later, I propose some indicators that measures the success of the webapp. Second section, explains how the webapp works at an users level and a technical level. Languages used the implemented code structure is comprised in this chapter. Finally, in the third section, data bases logic and testing approach is explained.

Conclusion

Application goals

It is known that the percentage of students who find a tutor by the date in which the deadline is close, presents a peak. Therefore, it is intuited that this large percentage of students did not know what topic of tutor suited them, otherwise they would have chosen a tutor before. The process of choosing a project topic and a tutor represents 50% of the planning process. That is from November of present year to March of next year. To tackle 50% of the entire dissertation process is a major issue, not just because of what represents out of the total schedule, but also, because it impacts the next 50% (from April to September). If students can engage in a dissertation topic that really interest them it can be argued that the quality could improve, and partial drafts are going to be delivered on time. So, the entire dissertation process is full of late applications, and dissertation topics which could be uninteresting for students.

Finding a topic is a one of the hardest things to do (Saunders, et al., 2019). Dissertation project selection website shows dozens of topics as well as more than 521 tutors (University of Warwick, 2022). On the other hand, students must select a dissertation topic when they have just taken 3 modules, being the first of them too general with no depth at the beginning of the course.

Research topic is supposed to be an excited phase (Saunders, et al., 2019) and not finding an interesting research topic negatively impact that expected outcome. The webapp must guarantee not only time and deadlines are accomplish but to stimulate students before and during their dissertation projects.

The webapp general objective is on facilitating the process of dissertation planning, from the very first step for students which is choosing the topic and tutor. I proposed the following indicators to correctly assess if students find the process easier (see table No. 1)

Table 1. Objectives and indicators

|  |  |  |
| --- | --- | --- |
| **Objective** | **Description** | **Indicator** |
| Time choosing topic | Amount of time (days) spend looking for a topic to research about | (Days spend with web app/Days spend without web app) -1 |
| Time choosing tutor | Amount of time (days) spend looking for a tutor to research about | (Days spend with web app/Days spend without web app) -1 |
| Philosophical approach on dissertation project | Correct understanding and usage of philosophical approaches on dissertation projects | (Survey results after webapp/survey results before webapp) – 1 |
| Students Satisfaction | Level of satisfaction of students who complete their dissertation project | (Survey results after webapp/survey results before webapp) – 1 |

As seen in table 1, particular objectives focus in some steps that are part of the process highlighted in the webpage (University of Warwick, 2022) such as finding a topic and finding a tutor, while other objectives focus on dissertation requirements such as philosophical paradigms (University of Warwick, 2022a) and finally a new quantitative objective is proposed based on satisfaction. This last one does not exist currently in University of Warwick.

The proposed webapp, intends to engage students in a topic and tutor that really interest them, and reduce the time spend in finding such subjects which at the end, can allow them to begin to research and writing before deadline.

Application front-end

I have described what makes hard the process of choosing a research topic and a tutor. Fortunately, I found creative solutions for similar problems such as choosing a career or master’s degree. For Instance, The Universities and Colleges Admissions Service (UCAS) (2022), has developed a quiz which guide students and prospects through a series of questions that helps them discover what career suits them. As seen in Figure No. 1, UCAS invites students to take the quiz, and later it shows 30 questions with the same 5 answers.

Figure 1. UCAS career quiz.

Text

Description automatically generated with low confidence Timeline

Description automatically generated with medium confidence

Under this logic, figure No. 2 explains the general logic and basic layout of the webapp based on a opencode done for a quiz (WebDevSimplified, 2019). By means of this logic and layout, the system will narrow student’s responses to help them know which topic is the most interesting for them and latter present tutors and other features explained later on this written work.

Figure 2. Basic webapp layout.

Chart

Description automatically generated with medium confidence A picture containing bar chart

Description automatically generated

A picture containing bar chart

Description automatically generated

As seen in Figure No. 2, the webapp narrows from broad topics seen in the Master’s in eBusiness Management such as Digital Marketing, Data or eCommerce. Other narrowing strategies can be done when asking if the student wish to do an entrepreneurial project. Each response is processed by the code and matched with a potential tutor. However, topics are too broad at this point. Figure No. 3 allows students to narrow to a second level of specialization. Depending on response made in the first level, a number of subtopics seen in those preferred modules are going to be seen in the webapp.

Figure 3. subtopics on modules.

A picture containing graphical user interface

Description automatically generated A picture containing bar chart

Description automatically generated

Finally, many tutors have sector experience or manage dissertation projects applied to specific sectors. This is the last stage proposed before showing the list of tutors to contact. The webapp will ask about the interest on certain sectors, with the possibility for students to not answer on those sectors questions.

Until this point, students should have a friendly and quick guide to discover their research topic and tutors. This means the webapp have covered 50% of the planning process. For the next stages, such as structuring the introduction, body and conclusion of dissertation project, I proposed the webapp keeps asking some issues important when writing. Some questions are as follow:

1. Do you enjoy researching based on reading literature and theories?
2. Do you enjoy researching based on interviews?
3. Do you enjoy researching based on numerical experiments?

As seen, all questions keep same closed-end responses structure of first stage of planning. Former question’s intention is to shred light into what kind of dissertation project is going to be. Whether is going to be a literature-based, interview-based, quantitative-based, among other. Shall not forget, the important think that the webapp solves is making decisions on what students are interesting in.

Same kind of questions with closed-end responses can be done to topics such as philosophical paradigms such as constructivism, positivism or other approaches.

Code structure

In order to develop this webapp, we have used three well known languages: HTML, CSS and JAVA. All of them proposed by WebDevSimplified (2019).

HTML structure was constructed to have two containers. One for questions and responses under the name of “Question-container”, and second one to contain control buttons such as “start” quiz and “next” question under the name “controls”. As show in Figure No. 3, HTML framework is very simple. Any modification on questions, answers or alternative text, will not be done in HTML code.

Although it is proposed 4 answers for each questions, there is no need to always have 4 answers. If less responses are proposed to a certain question, HTML code allows less questions without need of deleting any code in HTML.

Figure 3. HTML code

Links to CSS file called “style.css” and JAVA file called “script.js” are written at the beginning of the HTML code.

Both files were saved on a folder called “static” since we continuously test the output and looks of the webapp (as a strategy of development). For doing that we use a Flask library on Python. Such library states CSS and other file are saved in static folders.

Former paragraph acts on behalf on a continuous-testing approach explained later.

When not using Flask library on Python to deploy code, it is needed to erase “statis/” on line 7 and 8. And relocate such files on a desired folder.

Text

Description automatically generated

This code was originally built on a quiz logic, meaning that wrong and correct answers are shown. However, HTML code do not care about answers just yet. It is built to present the logic on questions, answers and a control button. As said by Patrick Cauldwell (2008) the important question about a webapp is not about building it entirely, but to figure out which part of the code should be developed inhouse and which part should be bought.

Other solutions such as google forms uses conditional surveys. It is also possible to built webapp through google sheets or forms. However, when analysing user experience and engagement of webapps, design is going to be highly important. Such scope is out of google forms or sheets.

Despite design and user experience is out of the scope, Figure No. 4 introduce a simple CSS code. As said before, the original code was a quiz with correct and bad answers. That is why most of the code structure is built on changing colours depending on correct or wrong answers. I have changed such parameters since there is no correct or wrong answer. Other codes on CSS file, refers to sizes, margins and locations of containers or text where questions and answers highlight.

Figure 4. CSS code.

Text

Description automatically generated Text

Description automatically generated

Text

Description automatically generated

Finally, and contraire of what can be done through a google forms, the original code proposed a JAVA script to stablish interaction. Such code is done by the same author of HTML and CSS code (WebDevSimplified, 2019). What the code should do is a follow-up, single page, design where students can easily scan and respond. Instead of writing questions within JAVA code, questions and responses must be in a database explained latter.

JAVA code is know as a “conditional branching” used for surveys, in which a sort of questions appear depending on responses of previous answers. To do so, the webapp could implement more complexity using JSON code or JAVA libraries as some resources suggests (YunaBraska, 2022; TechConative, 2020). However, there is no need to put so much complexity into the webapp. Specially, because JAVA language works under conditional statements easy to read and understand such as “IF STATEMENTS” or “ELSE IF”. Also, since students have 4 alternative answers (Strongly Disagree, Disagree, Agree and Strongly Agree), there are no unique result for students. Rather students have a range of tutors and topics ranking for the most popular to the less one. That means is easier at this point to extract responses, allocate a numerical value for each response and then using a simple mathematical equation to rank the best responses.

Table No. 1 explain former paragraph. Notice that each of 4 answers have a value between 0 to 3. Being 0 asigned to the response “Strongly Disagree) and 3 to the response “Strongly Agree). Also, notice that as said previously, I mentioned first level of questions related to the masters specializations (marketing, commerce, data science and digital transformation [not just consultancy]).

|  |  |  |  |
| --- | --- | --- | --- |
| Question\_id | Question | Answer | Mark |
| 1 | Is marketing one of your favourite Modules? | Strongly Disagree | 0 |
| 2 | Is Data Science on of your favourite Modules? | Disagree | 1 |
| 3 | Is ecommerce one of your favourite modules? | Agree | 2 |
| 4 | Is digital transformation consultancy one of your favourite modules? | Strongly Agree | 3 |

The reason to no mark “Disagree” with zero, is because I can interpret that there must be at least one subtopic related to that module that is interesting for the students. Otherwise, the student would have mark as “Strongly Disagree”. Thus, latter I could present some questions related to subtopics on that negative response to confirm if there is no interest at all in that module.

In the Figure No. 5, I have proposed a change in within the original code (WebDevSimplified, 2019) since we are facing a nested conditional questionnaire.

Application back-end

The webapp logic is a series of questions and responses that narrow interest in a module from the master’s degree, subtopics on chosen modules and research approaches and philosophical paradigms. To narrow every question based on former responses, it is proposed a series of data bases organized to show questions depending on answers.

Before talking about how data is structured, this webapp is basically a two-tier structure with an application used by an user, and a data system. However, and as trends and good practices mention, a three-tier or more is recommended for situations when a spike in traffic is expected (Laszewski, et al., 2018)

It is also important to notice that for practical purposes and to expose the concept of the webapp, we use a SQL data base that it is going to be explained in the next apart. Nevertheless, whether if SQL or NoSQL is needed in a further phase, is out of the scope of this written work.

On the other hand, the two-tier structure, and even the three tier structure, could be replaced in production phase for another structure that uses an “API gateway that uses an event-driven computing container as an endpoint, and a managed and scalable NoSQL” as Laszewski, et al. (2018) explained since it will allow the “team to focus on the business logic, and not how to achieve the scale required”. Former approach could be understood as a bottom-up processing because prioritize insights from data rather that organization of the data and is used for simpler data bases with few features or attributes (Connolly & Begg, 2015).

Other types of development that could be used on deploy phase, is single-page webapps since it allows webpages to load but not entirely which helps servers to run faster (Mozilla, 2022).

Data base structure

As said before this data base is done in a SQL structure, which have 8 tables: Students DB, Programs DB, Questions DB, Modules DB, SubModule DB and Tutor DB. Tutor Map and Module Map have the main goal to match responses with modules and tutor as students are narrowing and getting closer to their research topics. Other DB such as Sector DB or Philosophical Paradigm DB can be added later.

Figure 6. Relational Data Structure.

Diagram

Description automatically generated with low confidence

Figure No. 6 exhibits a relational data structure (Connolly & Begg, 2015) with low level of degree since each data base has no more than 4 attributes.

Each domain has been defined to be either of the following two:

1. Characters. Size 50, or

2. Integers. No size stablished.

Testing

Say about good practices

Conclusion

# References

**There are no sources in the current document.**