BIOCOMPUTING II

GROUP PROJECT

REFLECTIVE ESSAY-FRONT END LAYER

Name- Shaikh Mohammad Faizan

Email:faizanshaikh1897@gmail.com

May 2020

Birkbeck, University of London,

United Kingdom

Table of Contents

1.INTRODUCTION

Python and SQL are widely used programming languages with a wide range of applications. SQL is most common for database management systems.

HTML- Hyper Text Markup Language is a simple programming language made up of elements which describes the structure of a webpage. Elements are represented by tags and these tags label pieces of content such as “heading”, “paragraph”, “table” and so on. On the other hand, the Common Gateway Interface (CGI) is not a programming language but a method used by web servers to run external programs. The CGI script is usually called upon whenever a web page queries a database, or a user submits a form. JavaScript is a cross-platform, object-oriented programming language. It is used to make webpages interactive and add more functionality.

A Genome Browser which provides information on Human Chromosome 22 has been developed in a collaborative project in three layers namely, the Data Base layer, Business Logic layer and the Front-End layer. The database was created using SQL and Python. The Business Logic layer extracted the information from the database with Python through a user input query from the Front End. The Front End was responsible to display the data to the user with the help of CGI scripts.

2.APPROACH TO THE PROJECT

To complete this collaborative project, a team of four members was appointed wherein I was the fourth member. Being the fourth member, I was already assigned the task to carry out in the project by the supervisor which included two options: a) Develop an alternative front end
b) Create a detailed test suite and an automated set of code documentation. Front end development was a new and unexplored aspect for me; therefore, I chose the same as my task in this project. Initially, the different layers of the project were assigned to the team members based on their confidence, curiosity towards a new aspect or skill development. All the team members were happy with the work assigned.

2.1 INTERACTION WITH THE TEAM

The team interacted face to face either in meetings or group calls. The interactions were frequent in early stages but decreased later due to the unfortunate circumstances. It was clear by the third week about the final idea, requirements and individual responsibilities in the project and so the team members focused accordingly.

2.2 OVERALL PROJECT REQUIREMENTS

The approach taken by a programmer to perform a particular task depends on different aspects such as personal thinking, personal interpretation etc. However, the major requirements in the different layers were

a) Database Layer- Create a database and parse the GenBank file
b) Business Logic- Mid-point of communication between DB and FE layer, create functions to send and receive data and queries amongst different layers
c) Front End Layer- Develop a Web Front End to display data ad make searches available through forms.

2.3 REQUIREMENTS FOR MY CONTRIBUTION

As a front-end developer, my major task was to take data from the middle layer and format it for presentation. I received a set of functions embedded together in a module from the middle tier. I wrote CGI scripts for each function to interact between the layers and to send queries and receive data. The HTML-CSS scripts were used to design and develop the web front end.

3. PERFORMANCE OF THE DEVELOPMENT CYCLE

The overall performance in the development stages was inconsistent due to the unfortunate circumstances. However, as soon as every individual realised the responsibility bestowed upon him/her, everyone actively put efforts in completing the task. The interactions reduced gradually because of COVID-19 situation but the team was active on individual bases.

4. THE DEVELOPMENT PROCESS

In all the three layers, the ideas and suggestions on how to work on the project were welcomed from all the members. Although every individual had a different task, each member would contribute at least a suggestion to other members, if not much.

In the initial stages of the project, our team focused on creating the database and parsing the GenBank file into the database. This was a complex and a time-consuming task. The team discussed the idea of sending the database to the business layer as Python pandas dataframe, which would return a row or a value through a search input from the front end. However, this idea was neglected as the team realised an update in the database would cause the business layer to send out of date information to the front end. Several other ideas were discussed before settling on our current system, which takes user input query from the front end to the database via the business layer and returns raw information for that row based on the query. API's were created and team member working on the second layer started writing functions and ensured they worked correctly. Each function performed a different task and all the functions were embedded in a module to make things easy for the front-end layer.

As a member working on the Front-end layer, it was my responsibility to develop the Genome Browser. Most of the code required for the browser was scripted in HTML and CSS. The HTML was used to develop the browser while CSS provides style to it. The browser consists of nine different tabs including separate tabs for searching information on chromosome, restriction enzymes and codons present in chromosome 22. A glossary tab provides definitions of some basic terms used in the browser. Searches for user input queries in the browser are made available through different forms. The forms function with the help of CGI scripts, taking user input queries to the database and returning data via the business layer. Each form uses a separate function and all functions have their own CGI scripts.

5. CODE TESTING

The code testing for the first two layers was done using dummy data and the HTML-CSS scripts were tested in a dummy webpage. The CGI scripts were tested in the NoMachine platform using the Linux command line. All the CGI scripts were made executable before testing for their correct functionality. There were multiple iterations made and errors encountered during the process. However, it was ensured that all functions within the CGI scripts were able to retrieve the required data.

6. KNOWN ISSUES

It is good to minimize differences and issues while implementing a User Interface to make the UI predictable. Initially, the idea was to use JavaScript and AJAX for the front end development. However, as things progressed, there were multiple errors encountered and it became more difficult to debug the code. It was difficult to make it concise and up to the mark. Instead, CGI scripts worked better. There was an issue in deciding whether to call a function within the script or write a script for the whole function, which was settled along the way with better understanding of the scripts.

7. PROBLEMS AND SOLUTIONS

Under the current circumstances, it was quite difficult to manage the group project. The team was unable to interact in person. The team members found it difficult to explain the code to each other, however, the members shared videos within the group explaining each other about the processes and functioning of the code. The team also struggled with git commands due to lack of knowledge and less interactions.

Personally, I received several internal server errors even though the CGI scripts were written correctly. The same scripts when tested in the Linux command line showed no errors. These CGI scripts were rewritten going through multiple iterations. The CGI scripts were correct when tested in the command line, but the server displayed same error messages. A lot of time was lost in solving this problem. Moreover, the last three tabs didn’t display their contents when written in the same HTML document. Their code was scripted in separate files and each of the files were

called in the main HTML script with the help of <link> tag. I have had problems in connecting to the NoMachine platform from home and the problem continues to disrupt things in between.

8. ALTERNATIVE STRATEGIES

Writing the same functions received from the Business layer as a JavaScript was another way to meet the requirements. The XML AJAX was also an option but I rejected it due to lack of confidence. Frameworks such as Flask and Django were considered before settling for the CGI scripts.

9. PERSONAL INSIGHTS

This collaborative project has helped me enhance my knowledge in programming. It has improved my knowledge in understanding SQL and Python languages. Encountering several errors and problems and being able to overcome them was in itself a satisfaction. Understanding problems, correcting errors and understanding the interaction of code between multiple levels is a valuable skill. This experience has certainly helped me in doing so. It taught me how important it is to work together as a team and interact more frequently en route achieving the same goal. I believe that the project was very well organised. I also believe that under different circumstances, the performance of the team would have been much more convincing.

10. ACKNOWLEDGEMENT

First and foremost, praises and thanks to God, the Almighty, for his showers of blessings throughout my project. I am over whelmed in all my humbleness and gratefulness to acknowledge my depth to my team members, Annabel Page and Kostas Pantelidis, without whom this would have been incomplete and impossible. I would like to express my special thanks of gratitude to my teachers, Andrew Martin and Adrian Shepherd for organising this module and giving me the opportunity to be a part of a wonderful team. My heartfelt thanks.