Comp 2021 Project: HKUST Course Analysis

Group: Trine

Member: Chan Nok Hin (20349103, nhchanaa@connect.ust.hk), Cheng Ho Kei (12219689, hkchengad@connect.ust.hk), Sze Ka Yu (20348496, kyszeaa@connect.ust.hk)

## Motivation:

We found that a lot of time is wasted in enrolling courses as students are unsure about which course they may able to get in. We want to improve this situation by analysing the chance for a specific registration time slot to register a specific course, mainly for popular course. Thus, students need not to worry about it because they can immediately determine whether they can successfully enroll the course or not when they receive their registration time slot. 

They HKUST Course Analyser shows the data visualization of course registration trends by using past data to predict which year students tends to take this course by looking at the registration date. Moreover, it provides extra information about the course like course description, common core, required course and so on. It also updates a real-time trend description until end of Add-Drop period.

## Process:

### Tools:

#### Server Side:

For server side, PHPStorm is used to create php script for web grabber and generating graphs



#### Client side:

For client side, Dreamweaver is implemented to create html and JavaScript for designing web page appearance.



#### Database management:

For database storage, SQLite is a relational database chosen to store all grabbed data.

## Test data creation:

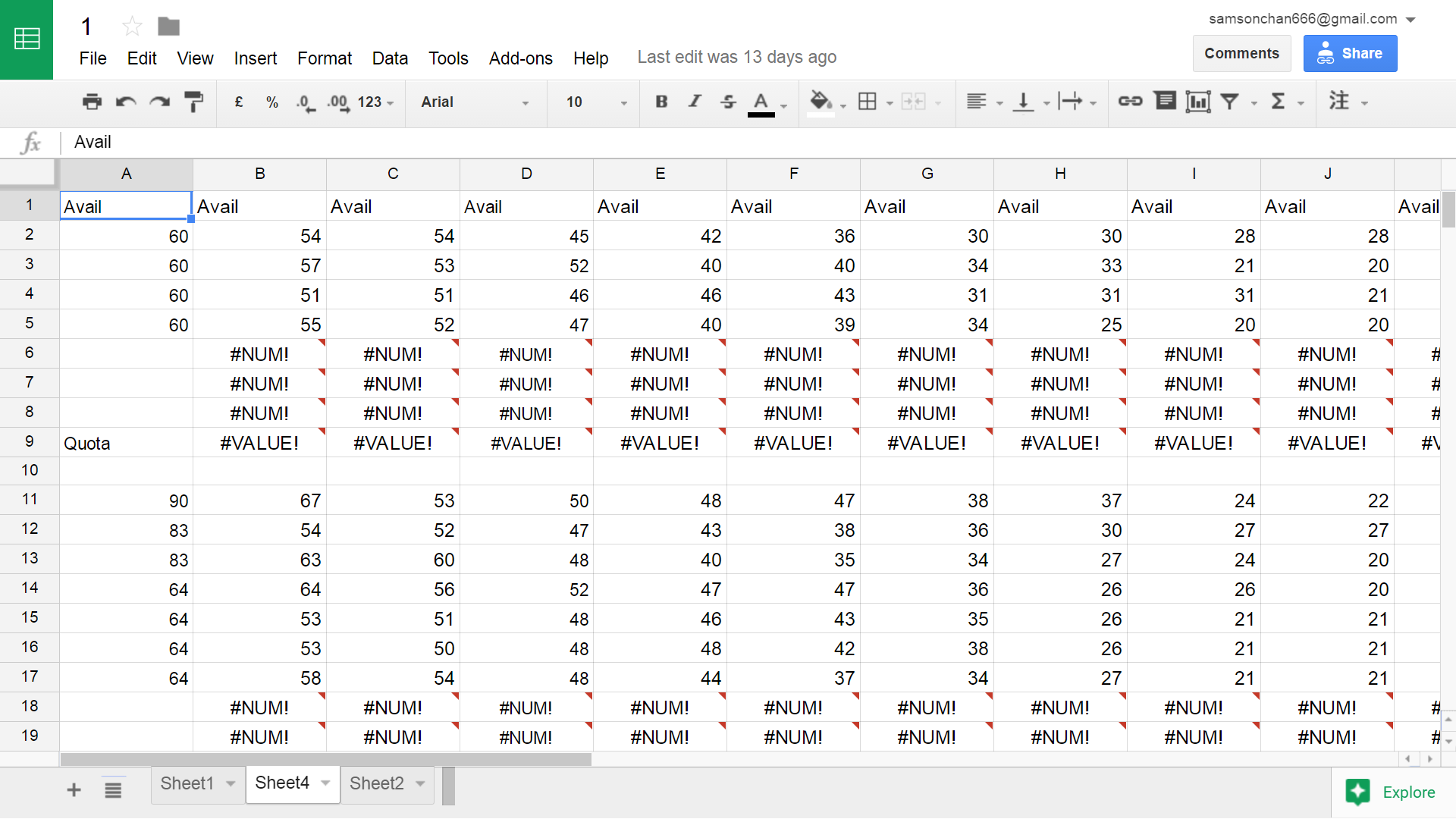


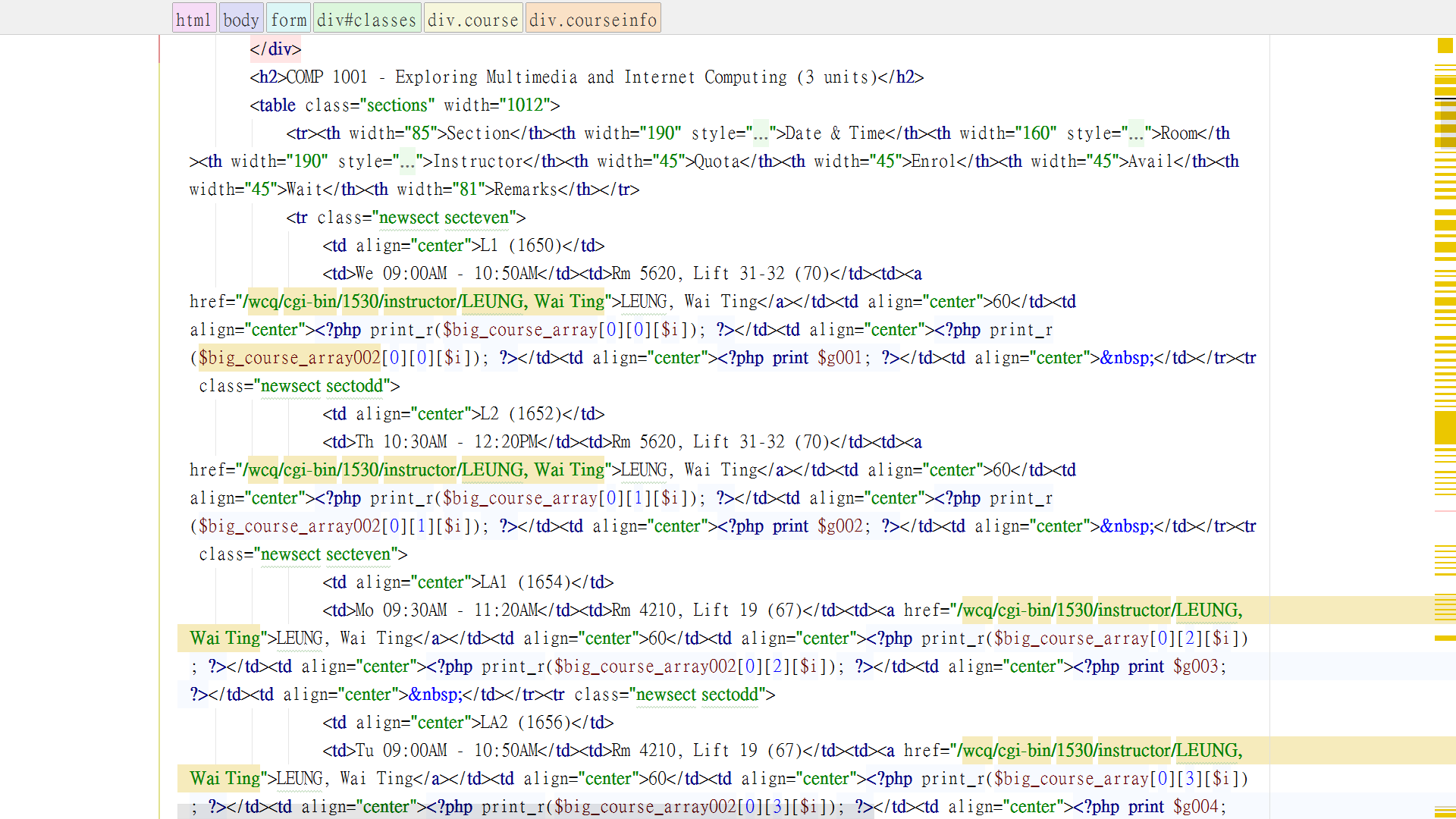
Excel is used to create datasets for testing and demo. PHPStorm is implemented to create PHP script for loading data from excel and constructing the test webpage. CSS is downloaded from “Class Schedule & Quota” and put into the test webpage such that it looks exactly the same as the source webpage.

## Result:

#### Test data and web page construction:

We have generated datasets for 2 years, each with 15 timeslots. Then, we used a PHP script to load data from the excel file into an html template which is exactly same as “Class Schedule & Quota” (with the same DOM and CSS).

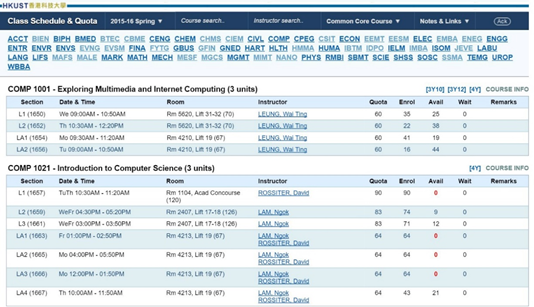


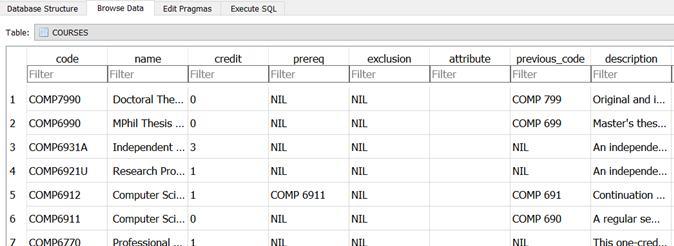


### Data Grabbing:

An automated gabber PHP script will visit the website “Class Schedule and Quota” and obtain its html contain. Then it would navigate in the html using DOM selector to obtain the plaintext, which is separated into individual items by regular expression and put into a class.

The obtained data will then be loaded into the database together with a timestamp, which can be used to plot graph by another PHP script.



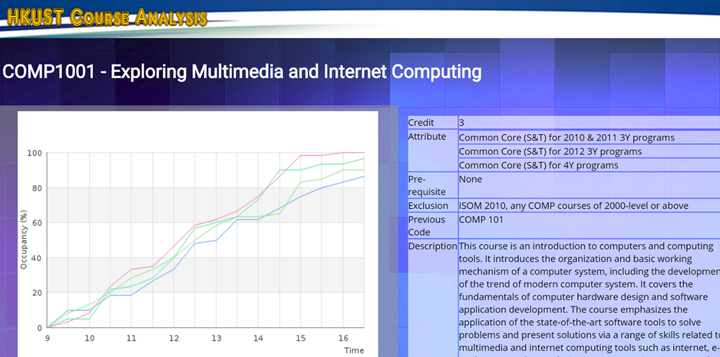


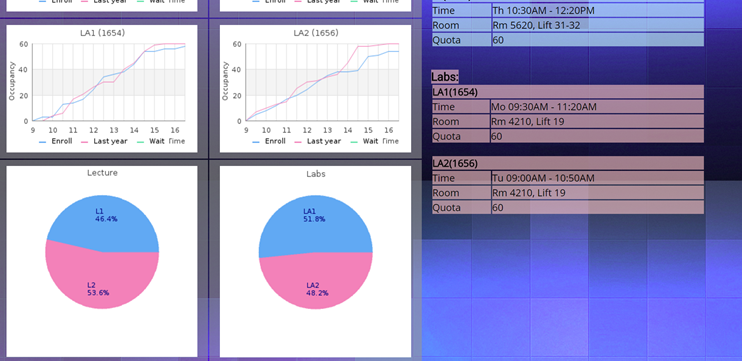
## Plotting graphs:

Different types of graphs are generated from the obtained data to analyse different results. For instance, line graphs give past and real time trend of the course enrolment and pie charts show the popularity of different lectures and labs to indicate which lecture or instructor is more popular.

Numerous sets of data are combined to output the most intuitive result so that students can have an immediate prediction and conclusion for which course they can or want to enroll in.

Extra information like credit, exclusion, description, lab time and lecture time is also shown on the webpage as a reference for students to have an overall and comprehensive view about the course.





## Challenges and solutions:

### Test data creation:

For test data creation, it is undoubtedly that constructing huge amount of course data for different time slots are very time-consuming.

Thus, excel is used to generate sensible time-changing data using specific formula, and data is packed orderly and input to the test webpage using PHP script.

### Data grabbing:

In data grabbing, it is found that entirely using regular expression is not suitable as there are too much cases to handle .It would be very hard to write and maintain an expression and we expect the code would run at a low efficiency.

Therefore, we imported a library “PHP Simple HTML DOM Parser” for DOM selection and used FireBug for finding out the DOM path in the html. Regular expression is only used in small scale for separation of plain text.

### Data Storage:

For data storage, several type of databases like MySQL and PostgreSQL have be considered. However, they are quite heavy and difficult to implement, given that our project does not focus on the database side.

At last, we decided to implement SQLite since it is fast and light. The only two disadvantages we found is that it does not support concurrent writing into the database or permission control, but we need neither of them.

### Graph Plotting:

For graph plotting, we need to look for a graph plotting library with high efficiency and caching function as we need to generate a lot of graph for every page visit.

We chose to use JpGraph to plot graph as it provide various kinds of graph and plot at a high efficiency. Besides, cache mode can be enabled for JpGraph, so that duplicate graphs need not to be generated when a lot of page visits happen in the same time which reduces server load to a large extent.