Department of Computer Science
The University of Hong Kong
Final Year Project
Final Report
The next generation eLearning platform for computing education

Supervisor: Dr. Chui Chun Kit
Student: Cai Tung San 3035090470
Group Members: Chan Wai Lun 3035089835
Cheung Siu Wai 2013588661
Wong Ching Quen 3035098161
Date of Submission: 16th April, 2016
Abstract

Regarding the loose linkage between e-learning platforms and computing education, this final report aims to introduce the brand-new e-learning platform which is specially designed for computing students.

There are already plenty of e-learning platforms which are available for schools and lecturers to choose. But when talking about current e-learning platforms for computing educations, these platforms do not have many aids that can reduce lecturers’ effort to monitor students’ performance as well as assist students on coding based on their current progress. Therefore, the project team would like to create a new platform which has functions to lessen lecturers’ workload and provide advice for students on their coding. Functions such as auto marking of code submitted by students, scoring of assignments can let lecturers focus on how the progress of the students instead of repeatedly marking of similar codes which may waste much time. On the other hand, students could receive relevant feedbacks that can guide them through their whole assignments.

The team has used C++ as the coding language of the code editor. Laravel framework is the web framework chosen due to its many power features and has good community power. Data analytics is the way that the platform mainly carried out to generate relevant feedbacks to students.

The rest of the report will include the deliverables, finishing status, difficulties on creating the platform as well as future planning of it for readers’ knowledge.

Acknowledgement

The platform would not be built possibly without the help of many individuals. One of them is our FYP supervisor, Dr. CK Chui. He has provided directions and scope of the platform and many useful suggestions to problems that we have encounter during the implementation process. We would like to give our sincere thanks to him.

On the other hand, the platform strongly relies on machine learning techniques to carry out data analytics. We would like to thank Dr. Andrew Ng and Dr. Li Yi Wei, which are the instructor of the Coursera online machine learning course and HKU machine learning courses respectively. Valuable knowledge of machine learning is gained from the courses taught by them.
Table of Content

1. Introduction..............................................................................................................5
2. Objectives..............................................................................................................6
3. Prerequisites..........................................................................................................6
4. Scope.....................................................................................................................6
5. Design....................................................................................................................7
   5.1 Front End........................................................................................................8
   5.2 Back End..........................................................................................................9
6. Platform Functions................................................................................................9
7. Code Editor...........................................................................................................18
8. AI Teacher...........................................................................................................24
   8.1 Classification of test cases..............................................................................24
   8.2 Ranking test case groups...............................................................................25
   8.3 Suggesting test cases groups........................................................................25
   8.4 Show example codes....................................................................................25
   8.5 Scoring of codes based on difficulties of test cases......................................26
9. Data Analytics.......................................................................................................26
   9.1 Data Preprocessing and Generate Correlation Matrices.............................26
   9.2 Learning...........................................................................................................28
   9.3 Evaluation.......................................................................................................30
   9.4 Creating the AI Teacher.................................................................................31
10. Example and Results............................................................................................32
11. Deliverables and Schedule................................................................................34
12. Difficulties encountered......................................................................................36
13. Final Status and Further Planning.................................................................36
14. Conclusion..........................................................................................................37
15. References.........................................................................................................37
16. Appendices.........................................................................................................38
List of Figures
Figure 1. Architectural diagram of the online e-learning platform
Figure 2a. Registration page
Figure 2b. Login page
Figure 3a. Profile settings page
Figure 3b. Change User Name page
Figure 3c. Change Email Address page
Figure 3d. Change Password page
Figure 4. Course Page
Figure 5a. Lecturer's view of Assignment Page
Figure 5b. Student's view of Assignment Page
Figure 5c. Lecturer's view of Files page
Figure 5d. Student's view of Files page
Figure 6a. Adding Test Cases
Figure 6b. Categorizing Test Cases
Figure 7a. Students' performance tracking
Figure 7b. Test case details
Figure 8a. Forum
Figure 8b. Quick reply under forum
Figure 9. Coding Page
Figure 10. Rename pop up window
Figure 11a. Console Window showing “Compiling”
Figure 11b. Console Window showing compilation errors
Figure 11c. Console Window showing output of the programming
Figure 12. Input pop up window
Figure 13. Console showing the input inserted from the student
Figure 14. Passing result of student’s submitted code
Figure 15a. Hint showed for the student
Figure 15b. Further hint showed for the student
Figure 16. Elbow method
Figure 17. Silhouette Plot
Figure 18. Different stages of the project
Figure 19. Submission matrices of 5 students

List of Tables
Table 1. Prerequisites of the e-learning platform
Table 2. Major differences for three chosen PHP frameworks
1. Introduction

Accessibility, user-friendliness, broad coverage of users and more are the most contributing factors, which make e-learning platforms into an important one in education field during recent decades. It appears to be unavoidable for students nowadays to use e-learning platforms like Moodle and i-Class in their study routine. Students can get access to the platform in anytime and anywhere to retrieve learning materials like lecture slides and readings, which are uploaded by the instructors. E-learning platforms can also serve as a medium for students to view, schedule and submit their assignments. Notice board and announcement system on the platforms can update students of the latest course information and arrangements in case of emergency. The platform thus can help students interact outside the classroom. Students can create new posts on the discussion session on the platform when they encounter any problem in their study without limitation of time or location.

In terms of the above basic functions, e-learning platforms seems to work very well in some concentrations. However, when it comes to computer education, the platforms appear to be not very capturing the needs of computing students.

For computing education, lecturers have to teach students different algorithms and data structures so that students can build up relevant computational knowledge for them to solve future problems. To know the understanding of students and let student comprehend these computational theories, instructors usually provide programming assignments for students to practice which students may need to apply the knowledge they learnt in the lessons to the assignments.

However, the linkage between e-learning platforms and programming appears to be weak currently. There appear not many tools from the platforms to facilitate the learning processes and to provide learning support to computing students. In present, the platform works with a single focus in assignments. When students work on their programing questions uploaded by their lecturer on the e-learning platform, they may not be able to code the problems on the platforms but have to use other code editors or even Integrated Development Environment (IDE) outside the platforms. After they submit the assignment back to the platforms, it takes time for the instructors to review and assess on the students’ work submitted before delivering feedback. The platform serves like a storage for assignments only, which is disconnected with students learning achievements and progress.

Spotting the problem, our team aims to provide a platform to fill this position. By aiding, students are expected to learn programming in a better way and lecturers could devote less time on review and asses of students’ coding assignments.
The remaining of this report will proceed as follows. First, to make the project direction significantly clear, we provide the objectives of what the project team would like to achieve by the platform. The report also lists out the prerequisites and define the scope of the project to indicate the boundaries the project. After that, the report will describe the design of the project in details to show the approaches that we have adopted to create the platform. Moreover, the report states the final status and further planning to illustrate what the project team have done in the platform as well what would like to do if the project proceeds. Finally, the report mentions the difficulties the project team have encountered and the possible solutions adopted to solve the problems so that the project can achieve current status.

2. Objectives

a. Allow students to learn programming with minimal efforts from lecturers which can lessen lecturers’ burden.
b. Providing a collaborative and CS specialized environment for computing students.
c. Provide timely coding feedback to students when doing their assignments based on their progress.

3. Prerequisites

Refer to Table 1 below, it shows the prerequisites which the project needs in order to build up the platforms.

<table>
<thead>
<tr>
<th>Medium</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server, virtual machine</td>
<td>- To compile the codes submitted by students</td>
</tr>
<tr>
<td></td>
<td>- To carry out data analytics</td>
</tr>
<tr>
<td>PHP framework (Laravel)</td>
<td>To act as a base of the e-learning platform</td>
</tr>
<tr>
<td>Data analytics’ tools and libraries</td>
<td>To act as a tools to carry out machine learning</td>
</tr>
<tr>
<td>(Scikit-learn¹, Numpy²)</td>
<td></td>
</tr>
</tbody>
</table>

*Table 1. Prerequisites of the e-learning platform*

4. Scope

The platform has code editor for students to write their programs for their assignments. C++ is adopted as the code editor supporting language as coding beginners are the target users of the platform and C++ is the first language computing students usually learn.

The data analytics by the platform concentrated on assessing student’s performance. Although the data in the platform could be used for data analytics, passing performance of students in test cases is designated as the first priority in the data analytics. Because it is more likely to produce useful outcomes which is highly related to students’ performance and achievements, instead of analyzing other data in the platform like posts on discussion forums, videos uploaded to the platform. These other data could be analyzed in the future progress of the project.

5. Design

Data analytics is the main element in the platform. The platform tries to analyze the test cases performance by students and generate feedback back to the students in the code editor. Detail of the platform are illustrated in Figure 1.

As shown in Figure 1, the platform is composed of two parts, the frontend which is in the middle of the diagram and the backend which is on the right of the diagram. Code editor and test cases adding by lecturers are carried in the frontend part, while the data analysis are carried out in the backend part. Below are the details of these two parts.
5.1 Frontend

The frontend part is a web system and user interface provided for students and lecturers to interact with the e-learning platform. For example, the frontend provides user interface for lecturers to post assignments and for students to view the assignments. Details of the functions in frontend will be described in Section 6.

PHP framework is used by the project team to facilitate the development of the system. Three popular PHP frameworks Laravel\(^3\), Symfony 2\(^4\) and Yii\(^5\) are the possible candidates. For all these PHP frameworks, they have critical library functions such as user access control and security support (CSRF protection) which can noticeably speed up the development of the platform.

All three PHP frameworks use traditional MVC (Model-View-Controller) structure for development. The Model part interacts with and retrieve information from the database. Laravel provides Eloquent ORM which allows us to perform model creation and query to the database easily. Moreover, the View part renders pages to users, all template systems allow us to extend and inherit layouts which make the viewed file clearer to understand. The Controller part handles the interaction between users and the platform. The Middleware and the inheritance allows us to assign action based on user’s interactions. This MVC structure establishes clear separation of duties and makes viewing and modification of codes much easier for other users.

The major differences are summarized in Table 2 below:

<table>
<thead>
<tr>
<th>Template system</th>
<th>Laravel</th>
<th>Symfony 2</th>
<th>Yii</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allow PHP usage in template system</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Online resources</td>
<td>Most</td>
<td>Middle</td>
<td>Middle</td>
</tr>
<tr>
<td>Extendibility</td>
<td>9000 packages</td>
<td>2830 bundles</td>
<td>2800 extensions</td>
</tr>
<tr>
<td>Logging management</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Table 2. Major differences for three chosen PHP frameworks

Laravel is the PHP framework chosen for the platforms’ web system. The major reason of adopting Laravel is the maturity of its online community on


offering interactive and timely support for developers. Thanks to the escalating popularity of Laravel, counterparts who encounter similar developing problems as us have heat up the discussion in the Laravel communities. Due to community efforts, solutions to problems are expected to be found easily that can secure the development process.

Thus, development time and difficulties could be reduced with the aids of Laravel community.

Another major reason is the extendibility of Laravel. Laravel provides more than 9000 packages, fast solutions for desired functions of the platform are expected to be found easily that can speed up the development process.

5.2 Backend

The backend part is a layer hide from the system that is used to store all the data created or uploaded by users. Data includes assignments, test cases provided by lecturers, forum posted by students as well as the passing results of test cases made by students’ submission. The backend also has compiler to compile the code submitted by students and returning the results. Furthermore, all the data analytics on passing result are carried out by data analytics to provide feedback to the students. Thus, the backend part is very important for the platform to provide the mentioned functions.

The platform use Ubuntu 16.04 as the operating system to run all platform code and to support the hosting of web system. MySQL and PHPMyAdmin are installed as to handle and store all the data in the database. To compile the programming codes submitted by students, the platform calls command line to execute, compile and run them.

6. Platform Functions

As mentioned, the frontend is used for users to interact with the system, thus many basic functions are provided and the details are listed below.

a. Login and Register as Users

For a new user to the system, he first has to sign up to start using the eLearning platform. The user is required to input his full name for later user personalization use, plus email address and password for future logins under the registration page. After the user clicks the “register” bottom, the system pops up a message to notify user the registration is successful. However, if there is error spotted in the filled information,
such as mismatched “password” and “confirm password”. The system will require user
to re-enter related fields before submitting again.

![Registration page](image)

**Figure 2a. Registration page**

Upon successful registration, system redirects the user to the login page shown in **Figure 2b**. Registered member of the system is required to enter their email address and password to sign in the system.

![Login page](image)

**Figure 2b. Login page**

**b. Profile Settings of Users**

A user can view and modify his user profile in the Profile Setting page shown in **Figure 3a**. Apart from his user name, belonged user group, registered email address and registration time, he can also commit actions including “Change User Name”, “Change Email Address” and “Change Password” upon request.

For example, the user wants to change a password for his account. He can click “Change Password” in **Figure 3a**. The system then directs him to the page shown in **Figure 3d**. The user can enter a new password the replace the old one. The change of password will take effect immediately.
Figure 3a. Profile settings page

Figure 3b. Change User Name page

Figure 3c. Change Email Address page
c. View Courses

Under the “Course” section, a user can view the list of courses he enrolled currently. As illustrated in Figure 4, the user is currently enrolled in the class of COMP 4321 *Introduction to Quantum*. The user can click a particular course and enters into the assignment page shown in Figure 5.

![Figure 4. Course Page](image)

d. Posting and Viewing of Assignments

Refer to Figure 5a & 5b, user can view a list of assignments in this page. Lecture can click the “Add” button to insert a roll for new assignments below current assignments. The grey circle shows lecturer relevant files uploaded in that assignment. Lecture can also delete selected assignment by clicking the trash bin icon on the right-hand side.

![Figure 5. Assignment Page](image)
Figure 5a. Lecturer’s view of Assignment Page

For the student’s view illustrated in Figure 5b, student can only view the list of assignments and a grey circle indicating with numbers of how many supporting documents of corresponding assignment have been uploaded by the lecturer. For example, the lecturer has uploaded one file in Assignment 1.

Figure 5b. Student’s view of Assignment Page

After clicking a particular assignment, the system redirects the user to the file page. Shown in Figure 5c, the upper part displays previously uploaded documents for the assignment. The lecturer can also delete any document by clicking the trash bin icon under action grid. He can also use the uploader section to add new files selected from his computer. Any duplicated files with same names will be overwritten by the system.

Figure 5c. Lecturer’s view of Files page

For Figure 5d, it demonstrates the student’s view of the files page. Student enters to the files page after selecting an assignment. He can retrieve the assignment materials uploaded by lecturer from this section. He can also know the file size and the last
modification date. To start doing the assignment, student can open the code editor by clicking “coding” under “Management”. The functions of code editor will be covered in.

![Figure 5d. Student’s view of Files page](image)

e. Adding Test Cases

Lecturers can add test cases for students to pass their programming assignments. A test case includes both input and output. Refer to Figure 6a, lecturers can add new test cases by clicking the “Add” button on the top. After clicking the button, a blank test case is added to the unknown group box, lecturers can type input and output in the textboxes provided, they can then click “Update” button to confirm adding the test case. If they want to withdraw a test case added to the assignment, they can click the “Delete” button.

![Figure 6a. Adding Test Cases](image)

f. Give Titles and Classified Test cases into groups

The platform can classify test cases into different groups. As shown in Figure 6a, there is 4 test cases groups classified as well as an unknown group of test cases. By clicking the “Open All” button on the top, the groups will open all and
show the test cases inside as shown in Figure 6b, when clicking “Close All” button, it will bring back to Figure 6a.

Lecturers can click the “Test” button for test cases in Unknown group, the platform will run through internal process and classified them to the test case groups. Lecturers can also give a title to the classified groups which those groups may correspond to special programming concepts.

Figure 6b. Categorizing Test Cases
g. View performance of students

Lecturer can keep track on the performance in an assignment of all students’ attempts in the page illustrated in Figure 7a. Students who have submitted their assignment would be arranged according their user ID in a list indicating their score. Referring to Figure 7b, lecturer can also check student’s performance under each test case. He can know student’s answer of the test case and compare it to the expected output. Lecturer can then check students’ understanding and identify some common mistakes made by students from the test case details.

![Figure 7a. Students’ performance tracking](image1)

![Figure 7b. Test case details](image2)
h. Forum posting questions

To construct a collaborative community in the e-learning platform, a forum is set up for students to raise and answer questions to assist each other’s learning. Student who has doubts on any related topics. He can create a new thread and post the question on the discussion board. Other students who know the answer can comment under the post. The forum is more beneficial and powerful than asking in class or individually. Concerning students may have similar questions, lecturer can save time from avoiding answering duplicate questions. Plus, student can check first the discussion forum before asking a question.

![Figure 8a. Forum](image)

After clicking into a thread, the post is marked as “read”. User can input his suggestion, opinion, or any possible solution in the quick reply section at the bottom of the page. The reply is not anonymous that user name is shown on each post. This is to maintain platform’s regularity and hold user’s replies responsible.

![Figure 8b. Quick reply under forum](image)
7. Code Editor

One of the objectives of the platform is providing a collaborative and CS specialized environment for computing students. To achieve this, the platform integrates external libraries or plugins to provide customized code editor for students to code together and use to submit assignments.

The platform embeds external code editor which grasp the advantage of code completeness and high degree of functionalities support. Firepad⁶, Ace⁷ and Firebase⁸ are the selected libraries for building our code editor.

a. Firepad
Firepad serves as real-time collaborative code editing tool. With firebase, students can code together just like using Google Drive File to type words. Students can see how other changes the code instantaneously. Thus, students are now easier to collaborate and code together, they can discuss how to modify their code and all the codes will be synchronized to other group members. They can also learn from others on how they change the code to become simpler or how others present the codes with the logic.

b. Ace
Ace serves as a common code editor which provides many coding general features. Examples are syntax highlighting of the keywords so that students are easy to recognize, code completion is supported that when students type some several characters, the code editor can suggest the whole words or variables so students do not need to type that many characters. Also, multiple cursors is supported that students can type multiple position at the same time, this is convenient when students have a lot of lines with words repeatedly. There are also much more details of features that Ace is supported which will greatly help the coding of students. The details of features are on Ace websites.

c. Firebase
Firebase serves as a native data logging system which records all input of students when they use the code editor. The records are stored and are greatly used in data analytics. Firebase stores all the changes including addition and deletion of codes so that the system can know how the students modify the code to achieve the passing of results. The use of Firebase minimized the time and workload needed for design and implement of data logging system.

Integrating these three libraries allow the platform to produce a good coding experience which can both improve their coding quality and reduce their time to code.

In addition to importing libraries, the platform has also provided other features that may be convenient or can further help students on their coding. Figure 9 below shows the user interface of the code editor that there is buttons on the bottom of code editor which could help students. The functions will be explained one by one.

![Figure 9. Coding Page](image)

### a. Rename File

The bottom left of the page show the filename. As shown in Figure 9, the current filename is “test.cpp”, student can rename the file by clicking the “test.cpp” word. A small window will pop up as like Figure 10 which shows a textbox that allow student to type the new filename. By clicking the rename button, the filename will be changed.

![Figure 10. Rename pop up window](image)
b. **Download Code**

On the right side of the button to rename the file is the download button. Clicking the “Download” button allows students to download the source code they or their group members have typed. They might store it for their future use or share it to others. Alternatively, students can also click “Ctrl + s” to download and save the source code.

c. **Auto Saving Code**

It can be notice that there is no saving button in the function bar of the coding page. It is to prevent students losing code because they did not click save or their page suddenly shut down, the platform code editor will help them save the codes automatically without the needs for students to click any buttons. They do not have to worry about the problem of retyping all the source codes lost.

d. **Compile and Run Code**

On the right of the download button is the “compile and run” button. By clicking it, the platform will compile the source code in the server and then return back the results. If the compilation fails, the platform returns compilation errors to the students, if the compilation success, the platform returns the output produced by the code. As shown in Figure 11a, A console window is shown on the right side of the code editor page which here showing “Compiling …”, either the errors or the output returned from the server will show up to the console for students to view. In Figure 11b, the compilation errors is shown in the console while In Figure 11c, the output of the program which is “Hello World!!!” is shown in the console.

![Figure 11a. Console Window showing “Compiling”](image)
e. Console Button
There is a “Console” Button on the bottom bar which allows students to close or open back the Console of the code editor.

f. Insert Input
On the right of the “Console” Button, there is an input button. As assignment usually needs to produce different outputs based on different inputs, the platform provides the input button for students to test their code based on example inputs which may be supplied in the assignments.
By clicking the "Input Button", a small window is popup as shown in Figure 12. Students can type the input in the checkbox and click update to update the input of the program. Here, the student types in “12345678” as the input.

![Input pop up window](image)

*Figure 12. Input pop up window*

After adding input to the code editor, and then compile and run the code, in Figure 13 the console can return back the input type from the code which is “12345678”.

![Console showing the input inserted from the student](image)

*Figure 13. Console showing the input inserted from the student*

g. **Submit and Check**

Students can compile and run as many times of their programming as they want and with different inputs supplied. Once they think they have finished their coding assignments, they can submit it by click the “Submit and Check” button. The source code will now direct to the server and run using the test case inputs and outputs supplied by the lecturers for the assignments. After checking, the platform will return the passing results of the students’ code.

As shown in Figure 13, the student submitted the assignment. The result is he passed 3/10 of the test cases supplied by the lecturers. The console also provides the input, output and expected output each by each. Students can know clearly which test cases they have failed and can try to modify their codes to pass all the test cases.
h. **Hint Function**

Students may get stuck when doing the assignments. Thus, the platform provides hints which is the feedback to help them pass the test cases. Students can get the hints by clicking the “Hint” button, a window is pop up to show the suggested test cases that students can try to pass first. These test cases could be easier compared with other test cases. If students still get stuck, a second hint is also provided which is some example addition or deletion of codes that other students do to pass the test cases. Students can learn from the example and try to modify their own code. **Figure 15a** shows the hints provided to the student when he click the “Hint” button. **Figure 15b** shows the code changes provided to the student when he click the “More Recommendation” button.

**Figure 14. Passing result of student’s submitted code**

**Figure 15a. Hint showed for the student**
8. AI Teacher

Another objective of the platform is to allow students to learn programming with minimal efforts from lecturers which can lessen lecturers’ burden. To reduce lecturers’ efforts, the platform provided an AI teacher which tries to do some of the jobs previously lecturers should do. The AI teacher is used to provide coding support and feedbacks which is the hints mentioned above to students and calculating results for teachers. The features of AI teacher are listed below. Section 9 will provide the details on how to achieve these features though data analytics.

8.1 Classification of test cases

As mentioned in Section 6, the frontend part has interface for lecturers to input test cases. The platform also has a “Test” button on the new test cases. Classification of test cases is done backend by clicking the “Test” button.

Based on data analytics, the AI teacher can classify all the test cases inputted into different clusters, test cases in the same groups are likely to be testing similar programming concepts of the programming codes. The AI teacher can cluster test cases into n groups, meaning that the test cases can test the codes with n different levels.

When new test case is supplied, the AI teacher can recognize whether it belongs to any previous test case groups or it tests a new programming concepts. Have more test cases groups means the test cases can test the codes more comprehensively and can ensure students’ codes have less bugs.
With the AI teacher, lecturers only have to add new test cases and can check the students’ codes in a better way, lecturers use less efforts here.

8.2 Ranking Test case groups
Besides classifying test cases into different groups, the AI teacher can also rank these groups by their difficulty levels. Knowing which groups is of which difficulty allow lecturers to check the progress of students.

For example, if most of the students pass medium level test cases, but several students only pass easy level test cases, lecturers may know those students could have difficulties in doing the coding assignments or can devote more time in helping those students understand those high-level programming concepts.

Moreover, if most of the students pass the test cases of all levels, that may mean lecturer could try to increase the speed of his or her teaching, or can provide much harder assignments in the next time.

Having the ranking of test case groups allow lecturers know the progress of individual students as well as the progress of whole class, which enables him or her to know what can he or she do to improve students’ performance.

8.3 Suggesting Test cases groups
Besides helping lecturers to know the progress of students, the AI teacher also helps students on doing their coding assignments.

As mentioned in Section 6, the code editor provides hints to students when they get struck and do not know how to do the next steps. One of the hints is suggesting test cases groups to the students by the AI teacher.

When clicking “hint” button by the students, the AI teacher will look through the submission made by the students and try to find out what test cases they do not pass. The AI teacher will then select the test cases groups which will be suitable to the students for their next steps. Students when receiving this hint can focus the effort on these test cases and try to modify codes to pass them. Thus, the AI teacher is providing a code direction to the students.

8.4 Show example codes
Another hint provided by the AI teacher is to show example codes from other students. Receiving suggesting test cases for direction is good for students to know what they might miss in their codes, but sometimes the patterns or concepts brought from the test
cases are not easy to be grasped by the students. Thus, AI teacher provides further hints which is showing codes.

The AI teacher tries to look for other students’ codes that can pass the test cases suggested to the students. After the AI teacher find the codes, he tries to get what codes are modified so that those students can pass the suggested test cases. He then provides back some codes to the current students as a reference. Having some codes suggested by AI teacher, students may know more skills in code and could try to use the skills learnt from the example codes to modify their own assignments.

8.5 Scoring of codes based on difficulties of test cases

The AI teacher also provides a good scoring system for lecturers based on their actual performance. For example, student A passes few very difficult test cases but fail to pass some easy test cases by careless mistakes, while another student B passes all the easy test cases and does not know how to do others, but turn out as the easy test cases groups may have much more examples than other test cases groups, thus student B with less stronger abilities may score higher than student A.

The AI teacher will try to avoid this situation, he tries to impose higher score on more difficulties test cases and lower score on basic test cases, thus the score shown to the lecturers are showing the abilities of the students.

9. Data Analytics

Data Analytics is carried out to provide the features of AI teacher mentioned in Section 8. The platform uses the passing results of test cases made by students’ submission to do analysis.

The most important analytics direction is to cluster test cases into different groups. Machine learning is used for the analytics in the platform. The flow of analytics are illustrated below.

9.1 Data Preprocessing and Generate Correlation Matrices

Data is created when students submit their codes to the server. It includes the passing result of test cases by that submission, output from students’ code as well as the source code submitted.

A submission contains results of passing of all test cases, student will try to modify and submit more to pass more test cases, thus there can be many students, each giving many submissions, and each submission give all the test cases results. These large amounts of data are used to calculate correlation matrices.
Below will be the steps to generate correlation matrices used for data analysis.

a. Submission Vectors
Each submission by student are modelled as a submission vector. For example, in a submission that the student the first three test cases out of 10 total test cases, the submission vector will be \([1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]\) where 1 indicate the student pass the test case and 0 indicate the student fail to pass it.

b. Delta Vectors
Delta Vectors are calculated based on two consecutive submission vectors. For example, when there is two consecutive submission vectors \([1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]\) and \([1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]\), the delta vector is the XOR between them which is \([0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]\). The delta means the changes in passing or failing of test cases in between two submissions.

c. Correlation tuples
Correlation tuples are calculated from the delta vectors computed. Correlation tuples are generated between the combination of 1 and 1 in a delta vector. For example, with a delta vector \([1 \ 0 \ 1 \ 0 \ 1]\), #1, #3 and #5 are the test cases with 1 in the delta vector, the correlation tuples are thus combination of them: (#1, #3), (#1, #5) and (#3, #5).

Correlation tuples indicate there is close relationship between the two test cases, more tuples in different submission means the two test cases are likely to be in the same group.

Correlation tuples are used to update the correlation matrices by a function of \(0.8x-2\), where \(x\) is the number of 1 in that delta vectors, in the case \([1 \ 1 \ 1 \ 0 \ 0]\), \(x\) is 3, value will be 0.83-2=0.8, the value is used to update the correlation matrices by the correlation tuples. (#1, #2) represents update row 1 and column 2 in the correlation matrix.

d. Penalty tuples
Like correlation tuples, penalty tuples are also calculated from delta vectors computed. Correlation tuples are generated between the combination of 1 and 0 in a delta vector. For example, with a delta vector \([1 \ 0 \ 1]\), the correlation tuples: (#1, #2), (#2, #3).

Penalty tuples indicate there may be a weak relationship between the two test cases, more tuples in different submission means the two test cases are unlikely to be in the same group.
Penalty tuples are used to update the correlation matrices by a function of $0.9x - 2$, where $x$ is the number of $1$ in that delta vectors, in the case $[1 \ 1 \ 0 \ 0]$, $x$ is $3$, value will be $0.93 - 2 = 0.9$, the value is used to update the correlation matrices by the correlation tuples. (#1, #2) represents update row 1 and column 2 in the correlation matrix.

**e. Correlation Matrices**

A correlation matrix indicate relationship between two test cases. It is a matrix with each entry range from $0$ to $1$. For example, if entry in row $3$ and column $4$ is $0$, this means test case $3$ is very unlikely or simply not correlated with test case $4$; if entry in row $2$ and column $1$ is $1$, this means test case $1$ and test case $2$ are likely to be correlated or are perfectly correlated. Higher value in that entry indicate a higher degree of correlation between those two test cases.

The correlation matrix is initialized for all entries with zero and all diagonal entries with one. This means it assume all the test cases are uncorrelated with each other but perfectly correlated with themselves. Then it is updated by summing the correlation tuples and take the average in each entry. Finally, each entry is multiplied by the penalty tuples.

The correlation matrices calculated are used for our highest goal which is to classify test cases into different groups.

Besides passing results are modelled as described above, the source codes are also undergone preprocessing to provide hints to students. The codes are undergone the DIFF function which works like the “diff” command in Bash.

**9.2 Learning**

After preprocessing of all the data, they are then used for classification in machine learning. Several unsupervised learning algorithms are used for the clusterization, they are K-means, DBSCAN and agglomerative clustering, their advantages and disadvantages are discussed one by one.

**a. K-means**

K-means is efficient to be executed with a complexity of $O(ikn)$ where $i$ refers to the number of iterations that the learning process iterates and give the results, $k$ refer to the number of groups that are needed to classified and $n$ refer to the number of test cases.

But K-means is not very efficient to deal with outliers, K-means tries to put outliers inside a cluster which will decrease the test case groups average correlations producing less accurate results. K-means also has to specified the
number of groups needed to classify, which the test cases added by lecturers, are unable to know how many groups should be clustered. Other methods have to carried out to find the number of groups first before carry out K-means.

b. DBSCAN

DBSCAN stands for Density-Based Spatial Clustering of Applications with Noise. To train a DBSCAN algorithms, two parameters have to be given beforehand, they are eps (maximum distance between two samples for them to be considered as in the same neighborhood) and minimum of samples surrounding a core point.

To cluster a group, DBSCAN tries to find test case data point A, draw a circle with radius eps, if the number of other test case points in that circle is larger than the minimum of samples specified, point A is a core point. All the other points in the circle belongs to a group. The area is expanded or overlaps with more core points found. The area overlapped with each other are considered to be in a group.

An advantage for DBSCAN is that outliers are likely to be ignored because they will be separated from the core points. In other words, the test case results which is abnormal can be ignored by DBSCAN. Also, the shape of area is arbitrary and depends on situation, while K-means’s shape is circle.

However, the two parameters in DBSCAN are not easy to find which may require frequent looping to find the optimal two parameters.

c. Agglomerative clustering

Agglomerative clustering initializes all the test cases as individuals. With n test cases, there will be n clusters. Then it tries to connect close test cases first, thus the clusters are merged gradually until a termination condition. The termination condition could be when n clusters are merged and finally k number of test cases groups are clustered. The process can also be done reversely which is try to group all the test cases, then separate the groups into 2 clusters which minimize the variance, separation continuous until termination condition reached and k number of test groups are find.

Agglomerative clustering provides a hierarchical relationship between test cases, such as a test case group has two inner test case groups. But similar to K-means, other methods have to carried out to find the number of groups (termination condition) and agglomerative clustering has a complexity of about O(n^2) which could be much slower than K-means.
Comparing the three algorithms, the project team chose K-means as for the test case situation, several methods could be adopted to find the number of groups easily. Also, K-means computer faster than other algorithms which the platform sometimes requires a fast result and when scale up, algorithms with lower complexity could be much better.

9.3 Evaluation

As mentioned above, methods have to adopt to find the number of test case groups suitable for K-means. To get the optimal number and produce good clustering results, two methods are used.

a. Elbow method

Elbow methods tries different number of clusters and calculate the percentage of variant explained. The result is plotted to a graph with percentage of variant explained or distortions against number of clusters. For example, in Figure 16, the elbow point is 4 which the change of distortions decreases in a large extent, adding more number of clusters does not greatly improve the results.

![Elbow method graph](image)

*Figure 16. Elbow method*

b. Silhouette Analysis

Silhouette analysis measures the cohesive of cluster as well as how the cluster is separated. Clusters with similar silhouette coefficients are better which means high coherence between cluster and high separation of clusters from each other.
Figure 17 shows a good clustering result that the three groups with data points in each group achieve similar silhouette coefficients distribution. The red line represents the average silhouette coefficient.

9.4 Creating the AI Teacher

After all the machine learning process explained above, K-means with data studied is produced. New data from the passing result of students and from the inputs of new test cases by lecturers can be analyzed. And the features by AI teacher can be produced from these data analysis.

a. Classify test cases into groups

Classification is done using Elbow method, silhouette coefficient and K-means algorithms to produce good clustering results. Using the correlation matrices mentioned above and from the passing result of students keep updating the correlation matrices, the classification keep changing based on students’ performance and try to achieve better results with more data given.

b. Ranking test case groups

Ranking test case groups is done by analyzing the submission matrices by the students, checking the test cases which have passed by students and get the order of test cases passing suggested. Test cases that are passed by students in the early submission are in higher order that they will be passed first by other students.
Computing the order coefficient of each test case, the project team then group the order coefficient by the grouped suggested in classification and compute the average order score of each test case groups.

A lower score of test case groups mean they are with less difficulties and could be suggested to students.

c. Suggesting test cases groups

Knowing how to rank test case groups allows the AI teacher suggests test cases groups as hint to be done easily. The AI teacher look for the test cases that student does not pass, put them in groups by classification, and then take the test cases groups with lower ranking score.

d. Show example codes

The method to show example codes is based on the suggesting test cases groups to the student.

The AI teacher finds other students with codes pass the suggested test cases groups, using DIFF and keyword recognized techniques, the AI teacher can then show example codes which is further hints to the student.

e. Scoring of codes based on difficulties of test cases

To score codes based on difficulties, the AI teacher get the test cases passed by that students’ codes, then group the test cases in the classified groups, sum up them using the groups average score, lower average score for easier test cases and thus the scoring is based on difficulties of test cases, then the AI teacher map the final score to 100 which will be the total scoring.

10. Example and Results

To test the functionalities and analytics result of the e-learning platform, server examples codes and students’ accounts are created by the project team.

The project team also created a simple assignment. The assignment detail is as follows: Using C++ to create a program that can accept input of HKID without the check digit, the program will output the check digit from the HKID input, if the input is not a valid HKID, output “Error”. The calculation of HKID check digit is in Appendices (Section 16.1).
The project team insert 10 test cases to the assignment, they are:

<table>
<thead>
<tr>
<th>Test case ID</th>
<th>Input (HKID)</th>
<th>Output (Check Digit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A123456</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>B234567</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>D456789</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>C345678</td>
<td>A</td>
</tr>
<tr>
<td>5</td>
<td>H123457</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>G123456</td>
<td>A</td>
</tr>
<tr>
<td>7</td>
<td>1234567</td>
<td>Error</td>
</tr>
<tr>
<td>8</td>
<td>1A23456</td>
<td>Error</td>
</tr>
<tr>
<td>9</td>
<td>AB987654</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>AB987655</td>
<td>0</td>
</tr>
</tbody>
</table>

*Table 3. Test cases for example assignment*

The project team creates 5 accounts, and try to solve the problems by typing codes, submit the assignment, modify again and submit until passed all the test cases. The submission matrices of 5 students are in Appendices (Section 16.2). Having the submission matrices, the platform can use to compute the correlation matrices mentioned in Section 9.1.

The elbow found from these test cases is 4, which means they are classified into 4 test cases groups. The groups are G1 (TC1, TC2, TC3), G2 (TC4, TC5, TC6), G3 (TC7, TC8), G4 (TC9, TC10). It can be observer that the first 3 belongs to a group which test simple HKID with 1 English character; the next 3 test cases belong to another groups that need handling of special check digit (10 and 11) calculated and map them to A and 0; the next 2 test cases belong to a group which is Error input group; the last two is a group that have 2 English character. The classification of test cases into groups seems to provide reasonable clusters.

The ranking of test case groups ranks by G1, follow by G2, then G4 and finally G3, which means G1 is the easiest group of test cases and G3 is difficult group among others.
Having the classification and ranking of test case groups, the project team tried with new account and submissions and to get hints. The hints are provided same the expected one that when the student has not passed TC4, TC5, TC7 and TC9, the hint will go suggest TC4 and TC5 for the student to try which belongs G2 and is of less difficulties than G4 (TC9) and G3 (TC7). When pressing further hints, the platform can also find the codes from previous students that passed G2 and suggested back to students.

Thus, the project team considered the data analytics of data to produce an AI teacher works well.

11. Deliverables and Schedule

The project is divided into two parts:

(1) The e-learning platform that will allow lecturers and students to view and submit assignment as well as using the code editor on the platform.

(2) The data analytics part which classify test cases and create AI teacher to facilitates students learning.
Each stage is built on the completion of previous stages. For example, code editor implemented in stage 1 by the platform team has to be used in stage 2 to generate fake data by the data analytics team.

Also, testing of the whole platform in stage 3 by the two teams together requires the finishing of the tools in the platform as well as the algorithms implemented in the data analytics.

Finally, the project team analyzed the platform as a whole and improves the two parts respectively.
12. Difficulties encountered

Because of the limited time, and only one year to create an e-learning system, with large number of functionalities suggested before, the project team still not able to implement them all. Thus, the project team focused on building core functions like the front end platform, code editor as well as the AI teacher. The project team managed to finish the core functions by this effective prioritizing of functionalities.

To facilitate the building of platform and data analytics, the project accessed many libraries and programming tools. To import the better libraries to provide better platform is not easy as well as knowing how to integrate them together. The team had therefore spent time and making effort on trying to build prototype using the libraries to test whether they actually work. This let the platform does not have incomplete functionalities and lots of bugs because of integrating external libraries.

There are many machine learning algorithms present which can cluster the test cases in the project. It was difficult to choose a good algorithm that suit this project, and the choosing of algorithms greatly affect the hints or feedback give the AI teacher. Thus the team has tried to analyze each of them and also implemented them to test on the data in the platform. The team also seek advice from our supervisor when encountered problems.

In the data analytics part, high dimensional data is produced from the correlation matrices, n-dimensional data for n test cases. It is difficult to visualize the change of data and to modify our modelling which could affect the evaluation and learning of K-means. Thus, the project team tried to use small test cases first and also used dimensionality reduction algorithms to facilitate the machine learning process.

13. Final Status and Further Planning

The project group tries to achieve the objectives made in this project. The platform could lessen lecturers’ burden as described in Section 8. The platform can also provide hints to students in helping them do their assignments.

The project group can achieve these because the power of a group of four people. The group is separated into two small groups each with two people. And the two groups deal with the two parts described in Section 10.

Although the project group can get to the two objectives, there are still many things could be done to improve the overall platform. For example, the platform can add more collaborative tools for students to use such as drawing boards, group chat and source code linked to video presentation. Moreover, the hints and classification of test cases can be further improved to
provide more useful directions and linked to programming concepts directly without the help of lecturers, it may also generate auto test cases without the need for lecturers to enter new ones.

14. Conclusion

In order to help lecturers and students for computing education, the project team presents this new platform which can lessen lecturers’ burden, provide a CS environment and coding feedback to students based on their progress.

This report described why the project team would like to create this platform, design diagram, user interface of this new e-learning platform. The report also mentioned the methodology behind to build the AI teacher and how the data is modelling. Moreover, it stated the libraries and the functionalities present in the platform.

The future planning of the platform is to provide more collaborative tools as well as improving the data analytics and has a smarter AI teacher. It is hoped with the new platform presented and keep on improvement of the platform in future, a good collaborative and CS specialized e-learning environment could be provide to lecturers and students so that computing students can learn concepts and theories faster and have better programming skills.

15. References

16. Appendices

16.1 Calculation of HKID Check Digit

a. Start with single English Character (e.g. A123456)

Step 1. The English Character is mapped to their alphabetical order with A = 1, B = 2, … Z = 26, i.e. A123456 -> 1123456

Step 2. First number multiply by 8, second number multiply by 7, etc.

1123456 -> 1 * 8 + 1 * 7 + 2 * 6 + 3 * 5 + 4 * 4 + 5 * 3 + 6 * 2 = 85

Step 3. Divide the sum by 11 and get the remainder: 85 mod 11 = 8

Step 4. Subtract the number by 11: 11 – 8 = 3

Step 5: If the digit calculated is equal to 10, check digit = A, if the digit calculated is equal to check digit = 0, else just output the digit calculated: check digit = 3

b. Start with double English Character (e.g. AB987654)

Step 1. The English Character is mapped to their alphabetical order with A = 1, B = 2, … Z = 26, i.e. AB987654 -> 12987654

Step 2. First number multiply by 9, second number multiply by 8, etc.

12987654 -> 1 * 9 + 2 * 8 + 9 * 7 + 8 * 6 + 7 * 5 + 6 * 4 + 5 * 3 + 4 * 2 = 218

Step 3. Divide the sum by 11 and get the remainder: 218 mod 11 = 9

Step 4. Subtract the number by 11: 11 – 9 = 2

Step 5: If the digit calculated is equal to 10, check digit = A, if the digit calculated is equal to check digit = 0, else just output the digit calculated: check digit = 2

16.2 Submission Matrices of the 5 students

TC: Test case
S: Submission
0: Fail to pass the test case
1: Pass the test case
<table>
<thead>
<tr>
<th>Student 1</th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
<th>Student 2</th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
</tr>
</thead>
<tbody>
<tr>
<td>TC1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>TC1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>TC2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>TC2</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>TC3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>TC3</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>TC4</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>TC4</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>TC5</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>TC5</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>TC6</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>TC6</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>TC7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>TC7</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>TC8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>TC8</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>TC9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>TC9</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>TC10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>TC10</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Student 3</th>
<th>S1</th>
<th>S2</th>
<th>Student 4</th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
</tr>
</thead>
<tbody>
<tr>
<td>TC1</td>
<td>1</td>
<td>1</td>
<td>TC1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>TC2</td>
<td>1</td>
<td>1</td>
<td>TC2</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>TC3</td>
<td>1</td>
<td>1</td>
<td>TC3</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>TC4</td>
<td>1</td>
<td>1</td>
<td>TC4</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>TC5</td>
<td>1</td>
<td>1</td>
<td>TC5</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>TC6</td>
<td>1</td>
<td>1</td>
<td>TC6</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>TC7</td>
<td>0</td>
<td>1</td>
<td>TC7</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>TC8</td>
<td>0</td>
<td>1</td>
<td>TC8</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>TC9</td>
<td>0</td>
<td>1</td>
<td>TC9</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>TC10</td>
<td>0</td>
<td>1</td>
<td>TC10</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Student 5</th>
<th>S1</th>
<th>S2</th>
</tr>
</thead>
<tbody>
<tr>
<td>TC1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>TC2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>TC3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>TC4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>TC5</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>TC6</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>TC7</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>TC8</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>TC9</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>TC10</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Figure 19. Submission matrices of 5 students