UBCO, CMPS Department, Computer Science

LearnCoding – Adaptable, Expandable Learning Platform

Dr. Abdallah Mohamed (PL), Thomas Van De Crommenacker (main dev)

What is LearnCoding?

LearnCoding is a teaching and learning platform created with flexibility and extensibility in mind.

Designed to create content that is accessible to anyone regardless of field of study, level of expertise or what language you speak.

Our platform consists of Courses that cover broader concepts within the realm of computer science and programming. For example a course might cover Data Structures. Within the courses there are any number of Lessons that teach and test the student on parts of of that concept. For example the Data Structures module may contain lessons on Binary Trees or Linked Lists. All the lessons can be accessed at any time and in any order for easy access.

What makes it different?

Extensibility

Although initially a lesson may only contain basic content in English, for the Java programming language and using examples from Computer Science. Instructors are able to contribute to lessons with translations, expertise in other programming languages or disciplines or other relevant information. This allows the existing lessons to serve as repositories of information that naturally grow highly accessible and informative.

Customizability

Naturally, as the lessons grow to contain more options for the user to choose from, the lessons become highly customizable to fit any student's need regardless of spoken language, level of expertise in a specific programming language or their field of study.

What's Inside a Lesson?

The lessons themselves consist of various pages of content, known as topics. These topics are currently one of three types:

- Text (including images, videos, and code)
- Coding Quizzes
- MCQ and Short Answer Quizzes

All the topics can be easily navigated to using the menu in the sidebar. They are completely accessible at any point since the aim is to allow the user learn whatever they're interested in.

The pages themselves are created using blocks of content that can be further configured to change with given set of customization options.

Example Lesson

Above you can see the current layout of a lesson with two different customization option sets selected. English and Java on top and French Python below. The student can navigate to different topics in a lesson, or other lessons in the module using the sidebar on the far left. The dropdowns for the customization of the lesson are found at the top of the page.

Building Blocks

Text, Images and Videos

Text articles allow an instructor to explore the lesson's topic. This may include examples with code, case studies, videos and images, specific applications and more.

Quizzes

The Quizzes can either be Coding Quizzes or MCQ and Short answer. The Coding Quizzes verify the user's code using unit tests written by the instructor.

Code Blocks

Similar to text content, code content can be added to lessons using code blocks within the lesson editor that have a code editor. Code blocks can be used by the student to test code or visualize code execution, step by step

Tracking Progress and Badgr

As the user completes content in a Course the system will track their progress. In order to complete a module a user must complete all the content that actually tests their new knowledge, that is to say they must complete all the Quizzes in a module.

Upon completing all the basic quizzes in a course they will be awarded a Badgr OpenBadge certifying their accomplishment. Upon completing all the advanced quizzes they are awarded an advanced Badgr OpenBadge.

Case Study

The system was piloted in CS1 courses (700+ students). A questionnaire using the Likert scale was used to collect student feedback. Regressions analysis was used to determine relationships between customizability and reported understanding and interest. All indicated statistically significant positive relationships.

Example Regressions

Perceived customizability of prog-language v. understanding (p-val = 0.000336). Perceived customizability of programming language v. understanding (p-val = 3.73E-10).

Tech Stack

The application has a Laravel PHP backend, MySQL database, jQuery was used for the front end. User code is tested using the Judge0 API running in a docker container. All of which runs on a RHEL8 VM on UBC servers.

THE UNIVERSITY OF BRITISH COLUMBIA

Acknowledgements

Project is funded by Aspire-2040 Learning Transformations (ALT-2040) Fund and the IKBSAS initiative funds.