**Team Good Friends**

**Proposed Level: Project Gemini**

**Motivation**

For many people, programming sounds like a complex field and often being stereotyped as the field for “nerds”, “geeks”, and “professional-only”. Nowadays, to popularize the use of programming, many websites and applications were created to help beginners familiarize with designing logical algorithms as well as practicing writing pieces of code, which provide great help for prospecting programmers.

In spite of those efforts, a lot of websites and apps lack the required user-friendly interaction to attract more hesitant or fastidious learners. For example, some utilize grid-in answer boxes rather than multiple-choice questions, discouraging users who have less time and concentration for learning. Also, we can introduce a form of leaderboard to encourage users to promote active-learning, which will lead to achievements or prizes.

**Aim**

We want to create an app which will help users to get used to coding concepts and languages easier than all existing solutions. In addition, the app will include a PvP mode and (potentially) a leaderboard to encourage people to learn faster in the mean of challenging friends and other learners.

**User stories**

* As a beginner who have zero knowledge about programming, I want a platform which is friendly for newcomers.
* After going through some training in coding, I want to hone my skills in a more pro-active way, rather than just studying.
* As an aspiring programmer, I want to try out my ideas and discuss them with other colleagues and friends. In addition, I would like to seek for interesting problems to test my abilities in problem-solving and code writing.

**Scope of project**

1. **Abstract:**

A web app including three main sections:

* Classroom: User can do mini-quizzes to learn and memorize code fragments and rules for multiple languages. After certain lessons, they can do a test to finish that section and move on. In addition, a reference library will be given for newcomers to read. Offered languages: Java, Python, C++
* Battleground: For users to test their abilities against other users, using a timed leaderboard.
* Sandbox: Free sharing and creating playground for users to test their ideas and explore possibilities.

The app will include a database of lessons, questions, and user-control server. An achievement system maybe implemented.

1. **Features included for June:**

* Classroom: lessons, tests.
* Battleground (as described above).
* Content database (quizzes, reference lib, tests’ questions, etc.).
* User database & login protocols
* UI/UX design completion.

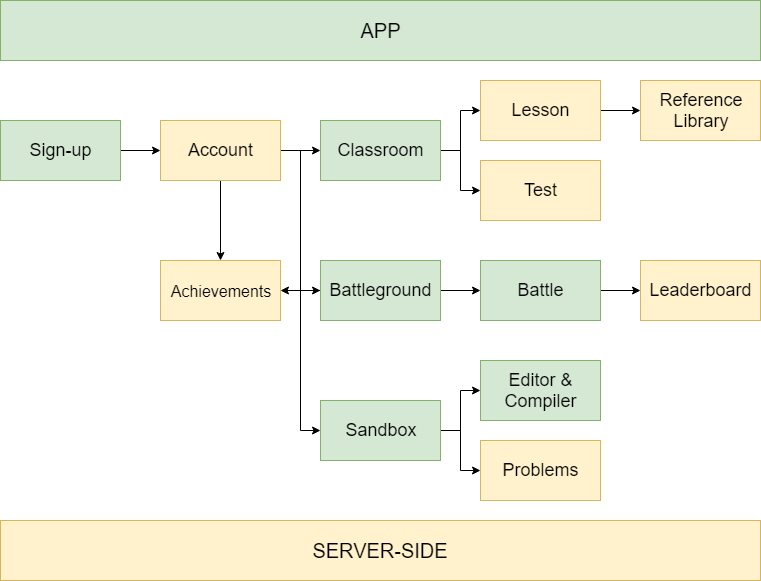
1. **Features included for July:**

* Expanded content database.
* Sandbox mode.
* Achievements and leaderboard for Battleground.

**Technologies**

* Backend: Python
* Front-end: HTML – CSS – JavaScript, Flask, React
* Deployment: Heroku
* Account authorization: Email, Google, Facebook

**Program flow**

****

**Comparison to similar existing platforms**

* SoloLearn:

+ Too heavy on pure knowledge, deterring newcomers.

+ Grid-in questions are not the optimal question type for beginners (in our opinion). Instead, we will utilize MCQs to create quicker quizzes and incentivize users to take guesses and memorize syntaxes & code fragments.

* W3Schools:

+ Simple design, detailed knowledge. However, their examples and quizzes are fairly easy and lack diversity.

+ There is no users battleground like SoloLearn.

+ There is no discussion forum/development ground for users.

* GitHub/StackOverflow:

+ Are not officially “learning platform”.

+ Do provide a great forum and knowledge storage, however, the questions and problems are too difficult and complex for beginners.

**Features as of Milestone 2:**

* A functional front-end app with two main features being ‘Classroom’ and ‘Sandbox’ (React):

+ ‘Classroom’: Three programming languages, each with several ‘levels of knowledge’ and tons of quizzes to complete and learn from.

+ ‘Sandbox’: A multi-language editor and compiler for users to try-out their ideas and getting rid of several editors for specific languages.

* Database of questions for ‘Classroom’ and ‘Battleground’. (MySQL)
* An authentication service for users to login and out of the application. (Python – Flask)

**Current difficulties:**

* We are currently having troubles with connecting our authentication with the actual app since the authentication was written with Flask, and the app itself was written with React.
* Currently the interface is still too simple (virtually no pictures and simple buttons). We will try to improve it to create a more game-like feel for the app in order to attract users.
* Our ‘Classroom’ is only providing quizzes for users to start with (similar to Duolingo). However, this is not good for newcomers because they lack the knowledge and skills to complete some of the questions. Therefore, we are working on a way to introduce a knowledge library into ‘Classroom’ to provide sufficient knowledge of each level the user is on.
* We are moving the ‘Battleground’ feature to the final phase due to having problems with compiling and comparing answers to the test cases in the database.
* We also spent tons of time learning about React and JavaScript, which both of us are not familiar with, thus a lot of the scheduling were postponed and pushed back to Milestone 3. We will try to reschedule our plan accordingly and properly with the current situation**.**

**Testing and debugging:**

* We have asked our Computing student friends from several schools to do some testing on the app and the questions in the database. Overall, testers gave good feedbacks on our ideas and execution of the quizzes. However, they also mentioned about the lack of UI and base knowledge for beginners, which are the parts we are trying to get it right in Milestone 3.
* We encountered a lot of bugs with connecting the app and authentication because the two parts were written on two different platforms, thus making it more tedious.

Overall, we have (somewhat) completed the main feature being a functional ‘Classroom’ for users to complete multiple-choice quizzes and learn from the answers. However, we need to address further a lot of problems, as mentioned above. The current state of the app will be presented in the video.

**Withdrawal from Apollo 11 to Project Gemini:**

Although we have tried to reschedule, but unfortunately due to having works during our internship periods as well as families’ businesses, we evaluated that our project will not reach sufficient requirements and testing for Apollo 11.

However, we have tried and refine all of the existing elements, in addition to implementing the rest of the planned features to reach sufficient Project Gemini requirements, which we will be talking extensively in the following section.

**Features as of Milestone 3:**

* Completed Battleground, where users will be pitched together in a 1v1 match to see who has a better knack in coding. The battlers will be given a coding problem. Whoever finishes it quicker and correct will be deemed the victor.
* Expanded database for Battleground and Classroom.
* Upgrades made for UI (adding demonstrative pictures, clean-up for main menu).
* Implementing login authentication to the main app. Users can now create new accounts within the app or login using Google accounts.

**Difficulties encountered:**

* Our main concern during the last phase was the integration of our login protocol with the actual app, which we didn’t completed in the previous phase, since the two components were written with Flask and React, respectively. However, we managed to combine it and it works smoothly now.
* We have to re-evaluate the method to rank users in Battleground. After collecting feedbacks and doing multiple testing on viability, we decided that the leaderboard style wasn’t the most optimal choice since users can copy code and paste it and call it a day. Instead, we are introducing a live 1v1 battle for user in order to test their skills on the spot (which is like Computing students’ PE).
* We managed our versions via GitHub and we faced several issues with updating the progress due to inexperience with the system. However, we managed to get used to it and completed the process fluently afterward.