

Revision document

Prepared By:

CL_FRI_1000_G19

Jayalath Danish Dassana

Nur Azmina Hannani binti Muhammad Shahrul Nizzam

Mahissh Pranav Surendhar Rajasudha

Chung Kai Shing Matthew

REVISION DOCUMENTATION

CHANGES INTRODUCED AFTER CODE REVIEW INCLUDE:

1)Progress bar has been added.

The progress bar helps user to continuously keep track of his progress.

2)Python playground implementation

This feature has been added where the user can freely practice his coding skills.

3)Signed out button

This makes it easier for the user to exit the web application.

4)Deadline feature

This ensures that quizzes are completed within due dates. It also tests if the student has grasped the concept taught in the modules. Quizzes after the deadline cannot be accessed.

5)Front end for quizzes

Students can now attempt user-friendly questions about the concepts taught.

6) Challenges

Students can access challenging questions on the modules learnt and test their understanding. They are even provided with the detailed solution if they do not know the answer by clicking on the “show solution” button.

6)Score and results summary

This allows the students to have a detailed summary of their quiz attempt.

7)Feedback for quizzes

This customised message allows students to improve and the next steps in their learning journey.

8) Download progress report button

This button will allow admins to download the report of specific users.

FEEDBACK RECEIVED FROM GROSS GROUP EVALUATION:

Note: *Only feedback which provided a useful output has been considered due to the other team being unfamiliar with Django framework used.*

Feedback 1:

File name: WelcomePage_views.py

Issue: Clarity

- L39, home_view()

The except clause's functionality is generalised to catch any error that occurs.

Suggestion: The implementation of error specific exceptions within the code is suggested as a best practice.

Changes introduced:

The exception clause has been removed.

The `if not request.user.is_staff and not profile_completed` ensures that only registered users with a completed profile, that is they have a student, parent or teacher status, can login.

```
18         if request.user.is_authenticated:
19             profile_completed = False
20             if hasattr(request.user, 'student'):
21                 profile = request.user.student
22                 profile_completed = profile.profile_completed
23             elif hasattr(request.user, 'parent'):
24                 profile = request.user.parent
25                 profile_completed = profile.profile_completed
26             elif hasattr(request.user, 'teacher'):
27                 profile_completed = True
28
29             if not request.user.is_staff and not profile_completed:
30                 return redirect('choose_user_type')
31
32 <>         return render(request, template_name: 'MenuPage.html', context: {'form': form})
33
34 <>     return render(request, template_name: 'WelcomePage.html', context: {'form': form})
```

Feedback 2:

File name: UserManagement_views.py

Issue: Clarity

L56, logoutUser(): The naming convention used for this function's handle is not consistent with the rest of the code. This function follows camel case whereas all other functions and methods follow snake case.

Suggestion: Rename the function handle to be in snake case format.

Changes introduced:

The function name now follows the correct snake case format as all other functions.

```
43  def logout_user(request):
44      storage = messages.get_messages(request)
45      for _ in storage:
46          pass
47      storage.used = True
48      logout(request)
49      return redirect('home')
50
```

Feedback 3:

File name: UserManagement_views.py

Issue: Clarity

- L122, L125 complete_profile()

Line 122 assigns a "parent_email" attribute to the user's instance. However, in line 125, a new variable "parent_email" is assigned to the same data, which is then used for validation in line 126-131. The creation of this variable is unnecessary as the parent's email can be referenced directly from the user's email attribute.

Suggestion: Remove line 125 and replace any reference to the "parent_email" variable with attribute "user.student.parent_email".

In doing so, this may also improve code readability, as the functionality of the validation code block is made more clear; in that it is comparing the parent's email to the user's "parent_email".

Changes introduced:

No changes have been made since we have decided to keep the variable name parent_email since it is a shorter variable name which makes the code more concise and easier to read.

Feedback 4:

File name: UserManagement_views.py

Issue: Clarity

- L34, L50, lecture_view()

There exists two functions with the exact same handle “lecture_view” defined in lines 35 and 50. It was observed that the only difference is they render either “lecture_page.html” or “submodules.html” in lines 42 and 57 respectively. This might cause a conflict when calling either function.

Suggestion: Rename one of the function handles to a distinct name, to differentiate their functionality (and potentially resolve a bug?)

Changes introduced:

The second function has been renamed to basic_module_menu view and have a different functionality now.

```
2 usages 1 matthewchungkaisheng
46 @login_required(login_url='/login/')
47 def basic_module_menu_view(request):
48     student = Student.objects.get(user=request.user)
49     progress_tracker = ProgressTracker.objects.get(student=student)
50
51     basic_module = LearningModule.objects.get(name="Basic Modules")
52
53     module_progress = ModuleProgress.objects.get(
54         progress_tracker=progress_tracker,
55         module=basic_module
56     )
57
58     context = {
59         'module_progress': module_progress
60     }
61 <> return render(request, template_name: 'BasicModulesPage.html', context)
62
```

Change requests:

FR 9 - Settings:

If the user selects this option, the program shall guide them to personalized options for sound, text size, etc and also allow access to different language options that the game is available in, which are English (Australia) and Malay.

Change Description: This functional requirement is requested to be removed.

Change Rationale:

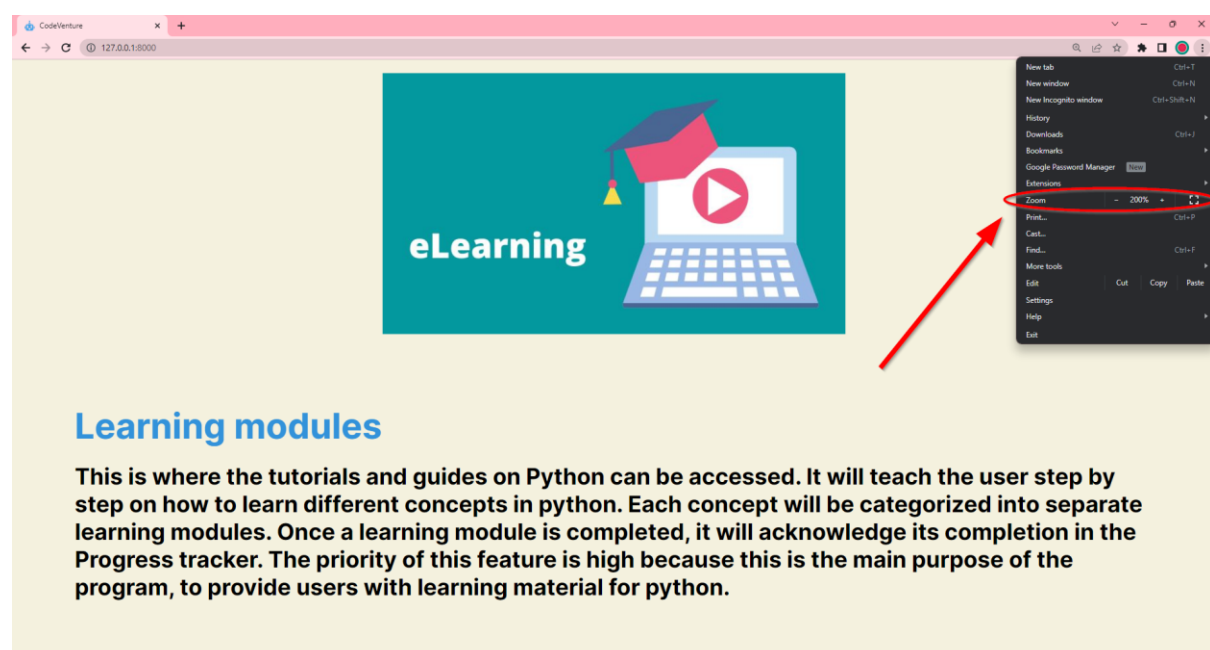
- All our videos are in English and therefore there is no need for game to be translated to Malay.
- Users can adjust the sound in the video itself, so there is not a need for a special button to adjust sound.
- Users can adjust the size of the text using the 'zoom' feature in the web browser. This will not hinder the user experience as the webpage is responsive.

Impact analysis:

While the settings feature could enhance the user experience to some extent, it is not a critical component for the core functionality of the system. Users can still interact with the system without these features. The change does not affect any other functionalities of the system and does not pose any security risks if not implemented.

Benefit of change:

Removing the settings feature simplifies the user interface and reduces unnecessary complexity, making the system easier to use. Moreover, relying on the web browser's zoom feature for text size adjustments takes advantage of existing browser functionality, reducing the need for a dedicated setting within the application.



FR 23 and FR 24 - Badge Collection and Badge Interaction:

If the user completes a challenge or a quiz, their award badges shall be displayed here.

If the user interacts with a badge, the program will display additional information about the achievement such as the associated challenge or quiz completed for that badge.

Change Description: These functional requirements are requested not to be implemented.

Change Rationale:

The primary purpose of the web application is to teach young learners Python. The introduction of a badge collection and information feature may distract young learners from their learning process.

By omitting the badge collection and information functionalities, a simpler and more user-friendly interface is maintained. Avoiding the complexities associated with badge management will make it easier for young learners to navigate the application and focus on their learning goals.

Impact analysis:

Excluding the "Badge Collection and Information" feature does not introduce any new security concerns, as it doesn't involve sensitive user data or major security implications.

The decision not to implement these features will not necessitate changes to the overall system architecture, as it does not introduce new complex functionalities.

Benefits of change:

The absence of the badge collection and information feature ensures that the application's purpose and objectives remain clear and consistent- teach Python, reinforcing the commitment to educational goals.

By allocating resources away from badge management, we can direct the resources towards enhancing the core educational content and features, leading to an improved learning experience.

FR 4: Leaderboard:

The leaderboard shall display the ranking between all signed up users according to their individual progress.

Change description: Only the front-end will be implemented.

Change Rationale:

The leaderboard feature, while valuable, is complex to implement. It would require significant development time and resources to accurately track and rank individual users based on their progress. Given the limited resources available, this feature is not feasible to implement at this time.

The primary focus of Code Venture is to provide a learning platform for young learners to understand Python programming. While a leaderboard can be engaging, it may also distract users from the learning content.

Implementing a leaderboard requires collecting and displaying user data related to their progress. This could raise privacy concerns among users. Not implementing the leaderboard simplifies data handling and ensures user data privacy.

Impact Analysis:

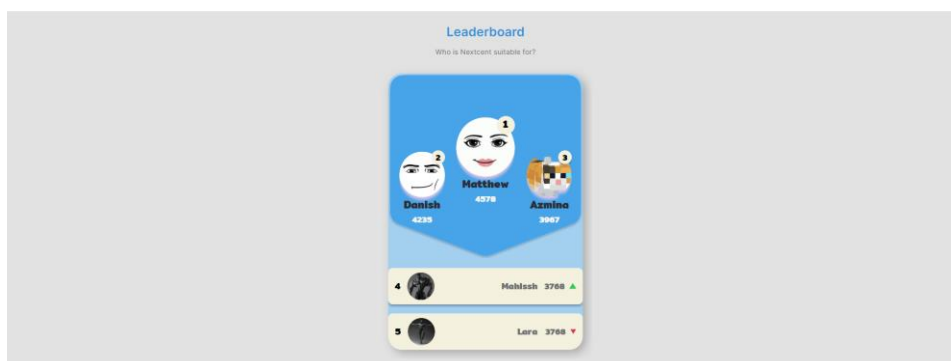
Not implementing the leaderboard frees up development resources and allows the team to focus on enhancing the core educational features of the application.

Benefits of Change:

By not implementing the leaderboard, development efforts can be channelled into improving and expanding the core educational content, making the learning experience more valuable for young learners.

User data privacy and security are maintained as we avoid the need to collect and display information on a leaderboard.

Users can fully immerse themselves in the learning process without distractions, ultimately contributing to a more effective learning experience.



- Only the front-end has been implemented for this functional requirement.