Requirement

Chào các em,

Cô gửi bài tập, các em làm xong, bỏ vào thư mục mang tên MaSV_HoTen rồi nén .zip gửi lại cho cô nhé.

- : Students should authenticate before accessing the system.
- :
- Username/Email
- Password

:

- Login button
- Forgot Password link

: Simple, clean, and secure login screen.

- : After login, students should land on a dashboard that displays available exercises to choose from.
- :
- : Display exercises grouped by programming language (C, Python, Java).
- :
- Exercise title
- Short description
- Difficulty level (Easy/Medium/Hard)
- Time limit (if any)
- Points or grades attached (optional)

: Allow students to filter exercises by programming language.: To search for exercises based on keywords.: Indicate whether an exercise has been completed, in progress, or not started.:

- Select an exercise
- View exercise details
- : Once an exercise is selected, students are taken to a coding environment to implement their solution.
- :

- : Display the exercise title and description at the top.
- :
- Syntax highlighting for C, Python, Java
- Line numbers and auto-complete
- Language switcher (to select the language)
- Pre-loaded template (based on the selected language)
- Instructions and constraints provided for the exercise

:

- Display predefined test cases with expected input/output
- Allow students to create and run their own test cases (optional)

:

- Run Code button to compile and run the code
- Clear Code button to reset the editor
- Submission button to submit the final solution for grading

:

- Run the code (compile + execute)
- Submit the code
- : After submission, students should see the results of their submission.
- :
- :
- Status (Success/Failed/Compilation Error/Time Limit Exceeded)
- Output: Show actual output versus expected output for each test case.

:

- List all test cases with:
- Input
- Expected output
- Actual output
- Status (Passed/Failed)

:

- Show score based on test cases passed
- If there are multiple test cases, show how many passed

: Display performance metrics like time taken and memory used (if applicable).:

- Try again button to return to the code editor for improvements
- Return to dashboard button
- : Show rankings based on student performance (optional, for competitive environments).
- : Allow students to leave feedback on exercises or ask for help.
- : Provide a hint button that can be used a limited number of times.
- : Allow students to save their progress and return later to complete the exercise.
- Code execution should happen in isolated environments (containers or sandboxes) for security reasons.
- Limit execution time to prevent infinite loops.
- Store previous attempts so students can review their submissions.
- : Each screen should have a simple design with minimal distractions.
- : Ensure the system is mobile-friendly so students can work from any device.
- : Clearly communicate syntax and runtime errors to students.