
Student Online Examination System

Overview:

The **Student Online Examination System** is a Python-based command-line application designed to simulate a basic exam environment. It allows users to add exam questions, administer a quiz to a student, and record the student's details and score in a CSV file. The system also supports displaying and managing questions, making it a versatile tool for creating simple exams and storing student results.

Features:

□ Question Management:

- o Add new questions to the exam.
- o Load existing questions from a CSV file.
- o Display, modify, or delete questions.

□ Exam Administration:

- o Administers an exam to a student. Collects the student's
- o name, university, and answers. Automatically calculates the
- o student's score at the end of the exam.

□ Result Recording:

- o Saves student name, university, score, and date of exam completion to a separate CSV file.

How to Use:

1. Running the Application:

- To start the application, run the “**question_services.py**” and “**examclinet.py**” file:
- You will be presented with a menu to either:
 - o Add new questions.
 - o Search a question based on question number
 - o Delete a question based on question number
 - o Delete all questions
 - o Modify a question based on question number
 - o Display all questions.
 - o Display all students.
 - o Exit

2. Administering an Exam:

- When administering the exam, the application will:
 - o Ask for the student’s name and university.
 - o Display questions one at a time and record answers.
 - o Calculate the final score at the end after submitting the exam.
 - o Save the student’s details and score to the student_results.csv file.

Code Explanation:

1. QuestionMaster Class: The QuestionMaster class manages the exam questions. It handles loading questions from questions.csv, adding new questions, modifying existing questions, and saving them back to the file. **Key Methods:**

- load_questions(): Loads existing questions from the CSV file.
- load_students(): Loads existing students from the CSV file

- ▢ `save_questions()`: Saves the current list of questions to the CSV file.
- ▢ `add_question()`: Adds a new question to the question list.
- ▢ `display_all_questions()`: Displays all existing questions.
- ▢ `delete_question_by_num()`: Deletes a question based on its number.
- ▢ `delete_all()`: Deletes all the question from the CSV file.
- ▢ `update_question()`: Updates the question based on its number.
- ▢ `display_all_students()`: Displays all the students from CSV file.

2. Exam Class:

The Exam class manages the exam process. It handles loading questions, starting the exam, collecting answers, and calculating the student's score. It also saves the student's results to `student_results.csv`.

Key Methods:

- ▢ `register_student()`: Registers the student.
- ▢ `start_exam()`: Administers the exam by showing questions to the student and collecting answers.
- ▢ `submit_exam()`: Tells the student that there exam has been submitted.
- ▢ `save_student_details()`: Saves the student's name, university, score, and completion date to `student_results.csv`.

Process Flow: I) FOR THE STUDENT

1. **Loading Questions:** Before the exam begins, questions are loaded from `questions.csv`.
2. **Student Info:** The student's name and university are collected.
3. **Asking Questions:** Each question is shown to the student, and their answers are recorded.
4. **Result Calculation:** At the end after the submission of the exam, the student's score is calculated and displayed.
5. **Saving Results:** The student's details and score are saved to `student_results.csv`.

Process Flow: I) FOR THE EXAM ADMIN

1. **Display All Questions:** Questions are first displayed to check if there are any existing questions.
2. **Add Questions :** Questions are added so that there are ready to asked to the student and are saved to the existing questions.csv file.
3. **Modify Questions:** Update the questions if needed using the question number.
4. **Display Question:** Verify the updated question using question number.
5. **Delete Question :** Delete the unwanted questions using their question numbers or all of them.
6. **Display students :** Load the students who have taken the exam.

Modules Used:

1. csv
2. os
3. logging
4. Question
5. Student
6. QuestionMaster