**MCQ-Based Online Exam Application Documentation**

**Overview**

The MCQ-Based Online Exam Application is a user-friendly platform designed to facilitate online examinations using multiple-choice questions (MCQs). This application comprises two main components: a Question Master for managing the questions and an Exam Client for conducting the exams.

**Components**

**1. Question Master (question\_master.py)**

The Question Master script is responsible for managing the database of questions. It allows users to perform various operations, including:

* **Add Questions**: Users can add new questions along with options, the correct answer, category, and difficulty level.
* **Edit Questions**: Users can modify existing questions, including changing options or other details.
* **Delete Questions**: Users can remove questions from the database.
* **Display Questions**: Users can view all available questions in the database.
* **Load/Save Questions**: Questions are stored in a CSV file (questions.csv) and can be loaded or saved as needed.

**2. Exam Client (exam\_client.py)**

The Exam Client script is designed for conducting the examination. It provides the following features:

* **Load Questions**: The script loads questions from the questions.csv file.
* **Conduct Exam**: Students can answer questions presented within a specified time limit.
* **Record Results**: The exam results are recorded in a JSON file (exam\_results.json), detailing the student's performance.

**Features**

* **Question Management**:
  + Users can manage a comprehensive database of questions categorized by subject and difficulty.
  + Supports multiple-choice format, with options for each question.
* **Examination Process**:
  + Conducts exams in a timed format, enhancing the authenticity of the assessment.
  + Allows students to review their answers post-exam.
* **Result Storage**:
  + Exam results are saved in a structured JSON format for easy access and analysis.
* **Logging**:
  + All actions performed in the Question Master script are logged in question\_master.log, providing an audit trail for all operations.

**How to Use**

**Setup**

1. Clone the repository or copy the files to your local machine.
2. Ensure Python 3.x is installed on your system.

**Running the Question Master**

To manage questions, run the following command:

* python question\_master.py

Follow the menu prompts to add, edit, delete, or display questions. All changes will be reflected in questions.csv.

**Conducting an Exam**

To take an exam, run:

* python exam\_client.py

Enter your name and university, set the duration for the exam, and answer the questions as they are presented. Results will be saved in exam\_results.json.

**Code Structure**

**Question Master Code Overview**

* **Class Definitions**:
  + **Question**: Represents a single question and its attributes.
* **Key Methods**:
  + load\_questions(): Loads questions from CSV.
  + add\_question(): Adds a new question.
  + edit\_question\_options(): Modifies options of an existing question.
  + save\_questions(): Writes updated questions back to CSV.

**Exam Client Code Overview**

* **Class Definition**:
  + **ExamClient**: Manages the exam process, including loading questions, conducting the exam, and saving results.
* **Key Methods**:
  + load\_questions(): Loads questions for the exam.
  + conduct\_exam(): Executes the exam flow and records responses.
  + save\_results(): Saves results to JSON.

**Logging**

All actions in the Question Master script are logged in question\_master.log, providing an audit trail for operations performed. This includes information about added, modified, and deleted questions.