Computer-Based Test (CBT) System

1. Introduction

This document serves as a comprehensive manual for the Computer-Based Test (CBT) system. It provides a detailed overview of the application's functionality and offers step-by-step instructions for both system administrators and test-takers. The system is a simple, lightweight web application designed to facilitate online examinations securely and efficiently.

2. System Overview

The CBT system is divided into two primary sections:

- **The Student Interface:** A user-friendly environment where students can input their registration number, access the test, and receive their results. The test is timed, and questions are randomized to ensure a fair testing environment.
- **The Admin Panel:** A secure, password-protected area for managing all test-related data. Administrators can create, edit, and delete questions, ensuring the test content is always up-to-date.

The application uses a server-side backend (Python with Flask) to manage data and serve the web pages. All user and question data is stored in local .json files, simplifying the setup process by eliminating the need for a separate database.

3. Administrator Manual

3.1. Initial Setup and Installation

1. **Project Files:** Ensure you have all the project files in a single directory. The core files are server.py, index.html, script.js, admin.html, and admin.js.
2. **Install Python:** If you don't have Python installed, download and install it from python.org.
3. **Install Flask:** Open your terminal or command prompt, navigate to the project directory, and run the following command to install the required Python libraries:
   ```
   pip install Flask Flask-CORS
   ```
4. **Data Files:** You must create two .json files in your project directory:
   - **users.json**: This file stores student registration numbers and names. The format must be a list of objects.
     ```
     [
       {"regNumber": "johndoe123", "name": "John Doe"},
       {"regNumber": "janedoe456", "name": "Jane Doe"}
     ]
     ```

- **questions.json**: This file stores all the test questions, options, and correct answers.

  ```
  [
    {
      "question": "What is the largest planet in our solar system?",
      "options": ["Earth", "Jupiter", "Mars", "Saturn"],
      "correctAnswerText": "Jupiter"
    }
  ]
  ```

5. **Start the Server:** In your terminal, run the main server file:

   ```
   python server.py
   ```

   The server will start, and you will see a message indicating the local URL (e.g., http://127.0.0.1:5000).

3.2. Managing Test Questions

1. **Access the Admin Panel:** In your web browser, navigate to http://127.0.0.1:5000/admin.
2. **Log In:** Enter the administrator credentials. By default, the username is admin and the password is admin. You can change these in the server.py file.
3. **Add a New Question:** Use the form at the top of the page. Fill in the question text, add at least two options, and specify the correct answer text. Click **"Add Question"** to save it.
4. **Edit an Existing Question:** Find the question you want to modify in the list below the form. Click the **"Edit"** button next to it. The question's data will populate the form at the top of the page. Make your changes and click **"Update Question"**.
5. **Delete a Question:** To remove a question, click the **"Delete"** button next to it. A confirmation prompt will appear before the question is permanently removed.
6. **Log Out:** Click the **"Logout"** button to end your session.

4. Student Manual

4.1. Taking the Test

1. **Access the Test Page:** In your web browser, go to the main page URL (e.g., http://127.0.0.1:5000).
2. **Enter Registration Number:** Input your unique registration number in the provided field. If the number is valid, your name will appear, and you can proceed.
3. **Start the Test:** Click the **"Start Test"** button. A countdown timer will begin, and the first question will be displayed.
4. **Answering Questions:**
   - Select an option by clicking on it. Your choice is automatically saved.
   - Use the **"Next"** and **"Previous"** buttons to navigate through the questions.

    - You can also use keyboard shortcuts: N for next question and P for previous question.
5. **Submitting the Test:**
    - When you've finished all the questions, or at any point you wish to submit, click the **"Submit Test"** button.
    - A confirmation prompt will appear to prevent accidental submission. You can also use the keyboard shortcut Y to bring up the submission prompt.
    - The test will also be submitted automatically when the timer runs out.

4.2. Viewing Results

- After submission, the test page will close, and a results page will appear.
- This page will show your final score and the total number of questions.

5. Troubleshooting

- **"Failed to load required data" error:** This usually means the Flask server is not running. Ensure you have run python server.py in your terminal.
- **"Invalid Registration Number" error:** Check that the registration number you entered is correct and exists in your users.json file.
- **"Page Not Found" error on /admin:** Make sure the server.py file is running and that the path is spelled correctly in the browser's address bar.