User Manual

[P2023-11] Automated Marking and Feedback System

Table of Contents

Introduction

Thank you for choosing Automated Marking and Feedback System (AMFS)! The system is developed to relieve the tedious job of marking programming assignments for instructors. AMFS is capable of completing the task by integrating three key components:

Automated Marking, ***Feedback Generation***, ***Plagiarism Detection***.

Outstanding features:

- **Interactive marking setup:** The settings of marking configurations are guided step by step, so users do not have to worry about building complicated python or shell scripts by hand. The system will start marking based on user inputs and handles the process for all submissions.
- **Tailored feedback design:** Separate feedback message corresponding to single test case can be submitted to the system with the test case files. AMFS also supports combinatorial feedback for submissions that fail specific combination of test cases within, providing more flexibility and freedom for the messages.
- **Feedback report render:** Users do not need to spend time designing the layout of the reports. The system generates all the feedback reports for you, containing all details you may take into account.
- **Integrated plagiarism detection:** Comparing similarity of all submissions is also a job of AMFS. The functionality is powered by MOSS (Measure Of Software Similarity) service from Stanford, and does the work silently in the background for you.

Requirements

1. Operating System

macOS 11 Big Sur or later.

2. Programming language and environment

The system currently only supports marking for Java-based code submissions. Hence, for successful marking procedure, users should have Java SE or JDK already installed on their machine.

3. Marking input and output

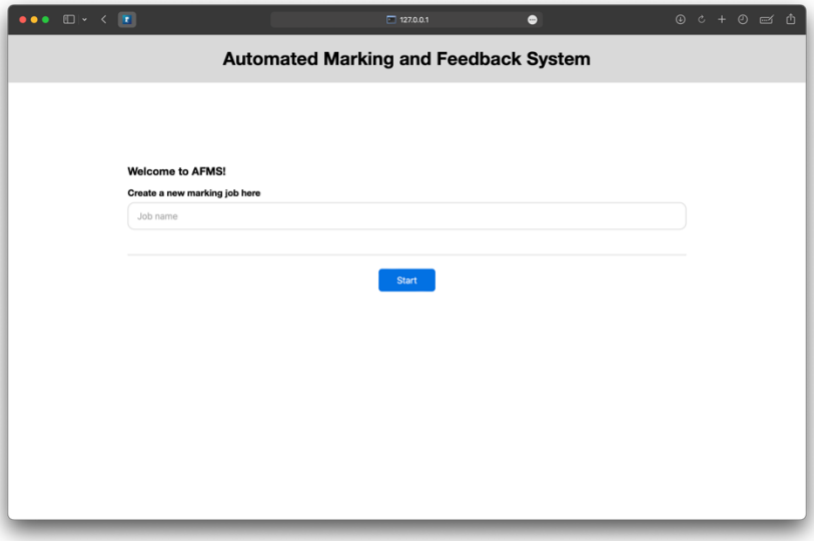
AFMS conducts marking by running subprocesses, sending inputs using `stdin`, and evaluates the score by comparing program outputs, `stdout`. There are some requirements for the marking assignments sent to the system.

- Users should supply an input file (`.in`) for each test case they would like to run while maintaining the execution command the same all the time. It currently does not support running tests by changing arguments in the command while marking.
- Programs to be marked should send output to `stdout` directly when running. The system cannot handle assignments that write output results to external files since the system now only captures program `stdout` in runtime.

Getting Started

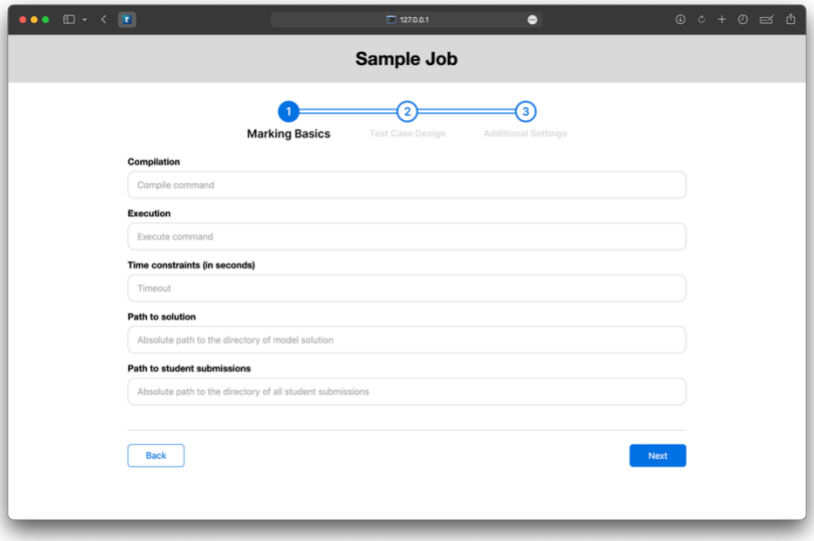
Welcome Page

(1) Create a name for the job you are about to conduct.
The name will also appear in the title of feedback reports.



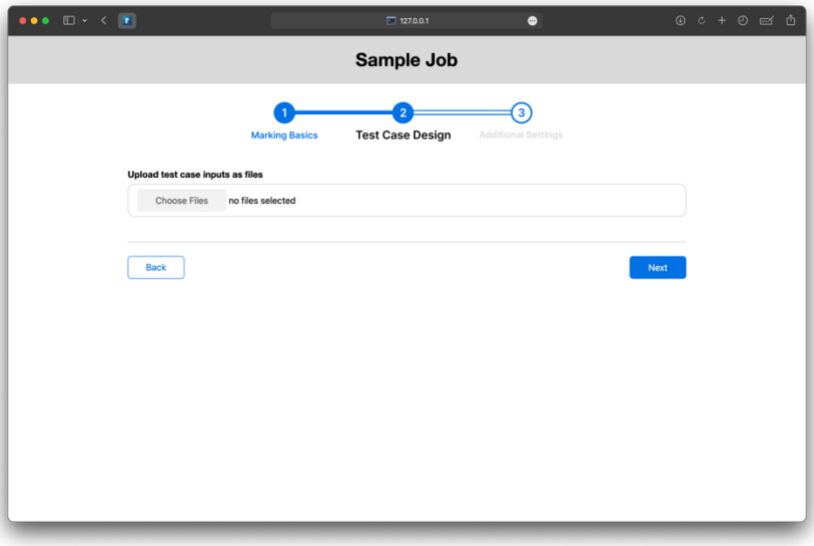
Setup – Basics Page

(1) Enter the command for compiling & executing the submission files as in the CLI.
(2) Enter the timeout limit for the execution. *You should enter an integer > 0.*
(3) Enter the **absolute path** to the directory containing the sample solution & subdirectories of student submissions.
You can use the `pwd` *command to generate the path.*



Setup – Test Case Design Page

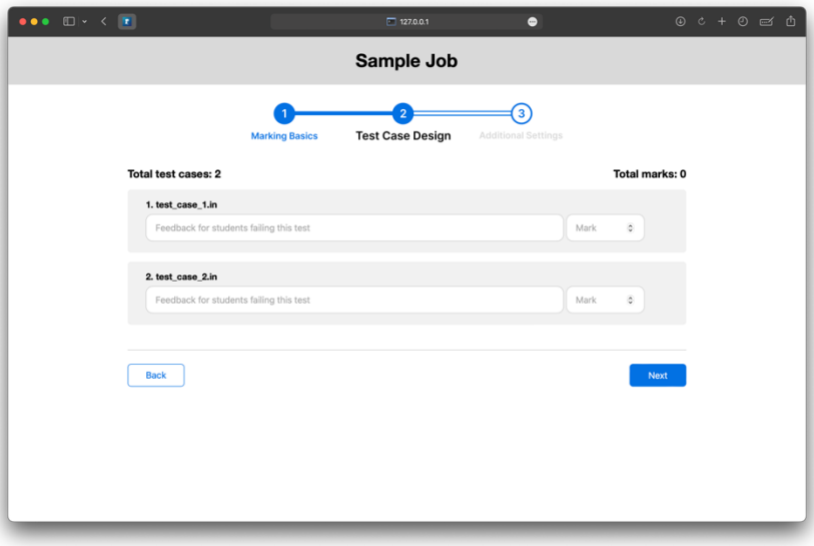
(1) Choose the input files (ideally, `.in` files) for each test case in the marking process, and the system will create a configuration block for each uploaded test case.



(2) Enter the feedback message for students failing this test.

(3) Enter the mark for this test.
You should enter a floating-point number ≥ 0 with one decimal place.

The system will help you sum up the full mark showing on the top-right corner.



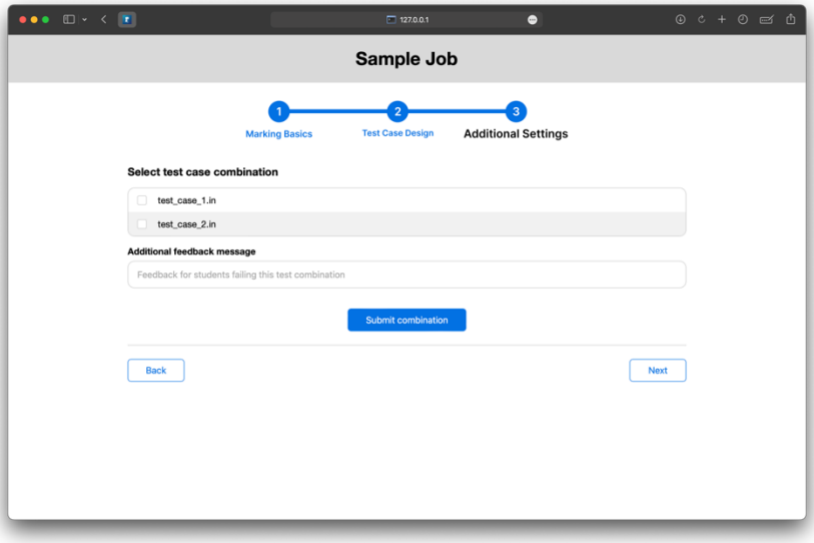
Setup – Additional Settings Page

This step is optional if you wish not to submit any additional feedback messages for students failing a combination of test cases.

(1) Check the test cases you want to write combinatorial feedback on.

(2) Enter the feedback message for students failing this combination of tests.

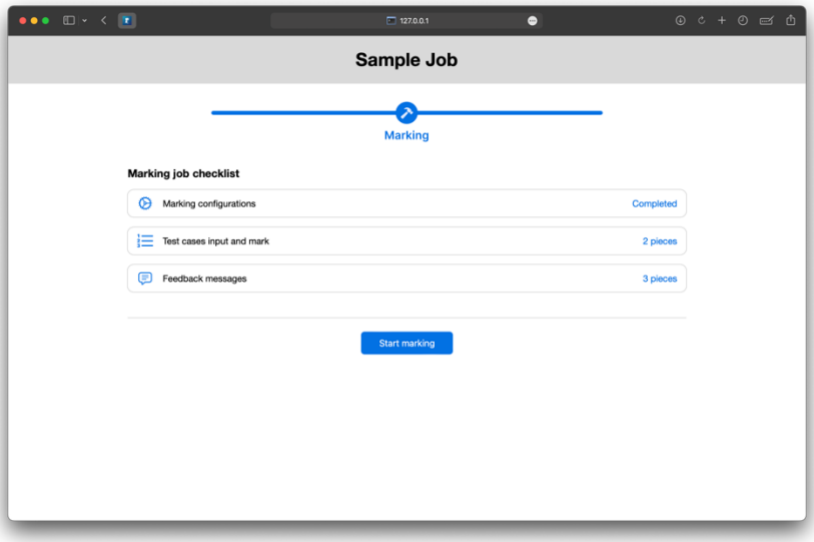
(3) Submit the additional feedback to the server.
You can only submit each combination only once to prevent repetition.



Marking Page

You can see a high-level overview of processing marking job, including the number of uploaded test cases and feedback messages. Click 'Start marking' to begin.

Please do not close the application window when marking, unexpected results might occur.



Results Page

Note that all the individual student feedback reports have already been written into the corresponding submission folder as PDFs, named as "`feedback.pdf`".

This page shows the results of the completed marking job:

(1) Marking statistics: AMFS calculates the average student performance as well as details including pass rate for each test case.

(2) Suspected Plagiarism: All student submissions are automatically uploaded to the MOSS server for plagiarism check.
The third-party service is not guaranteed to be working as expected all the time. There may be times when MOSS server is unavailable, you should try again later.

(3) As an instructor, you can download a snapshot of this results page by clicking the 'Download results' button so that you are able to review the statistics after you close the application.
The results are not preserved in the application, you are not able to recover it after quitting and have to mark again.