Opis funkcji controllerów w projekcie:

- moduł client\_controller.py:

* getClientId() – funkcja pobiera od serwera identyfikator opisujący klienta
* getUpdatedClientData(client\_id, source\_to\_compile) – funkcja wysyła request typu post do serwera wraz z danymi (klient\_id oraz przesłany kod źródłowy), zapytanie zwraca obrobione przez serwer dane
* show\_blank\_client\_view() – funkcja wyświetlająca podstawowy widok klienta
* parse\_message\_from\_server(input\_json) – funkcja pobierająca argument opisujący wiadomość przesyłaną pomiędzy klientem a serwerem w postaci JSONa, konwertuje go na obiekt pythona a następnie zwraca go z funkcji
* upload\_file() – funkcja obsługująca przesyłanie pliku do program klienta, po wczytaniu pliku jest wywoływana funkcja ‘getUpdatedClientData(client\_id, source\_to\_compile)’ z pobranymi argumentami. Następnie zostaje wykonane przekierowanie na funkcje wyświetlająca zaktualizowany widok klienta.
* show\_update\_client\_view() – funkcja wyświetlająca zaktualizowany widok klienta

- moduł server\_controller.py:

* verify\_program(file\_to\_compile) – funkcja sprawdzająca poprawność przesłanego kodu źródłowego I kompilująca przesłany program.
* parse\_message\_from\_client(input\_json) – funkcja parsuje argument przesłany w postaci JSONa I zamienia go na obiekt python.
* update\_client\_data(client\_data) – funkcja aktualizująca dane opisujące podłączonego klienta np. uzupełnia informacje czy dany program jest kompilowalny i outpucie programu
* show\_server\_view() – funkcja wyświetlająca widok serwera
* conectedClients() – API serwera, zwracające wszystkich podpiętych klientów
* sendNextClientId() – API serwera, zwracające unikatowe id klienta
* sendDataToClient() – API serwera, zwracające zaktualizowane dane o kliencie

Testy:

Napisano testy dla każdej zdefiniowanej metody w obydwu kontrolerach, czyli dla serwera i klienta.

Plików html nie przetestowano (ale to jest front-end,czyli widoki aplikacji) i chyba tego się nie testuje.