Test Case 1

Project Name: Automated Course Scheduling System

Test Case ID: 0101 Test Designed by: Akhat Suleimenov

Test Priority (Low/Medium/High): H Test Designed date: 26.11.21

Module Name: Import Course Information Test Executed by: Dariga Shokayeva

Test Title: Parsing data from working Albert Test Execution date: 26.11.21

Description: the python script for parsing data is run to get course information from working Albert

Objective: test if the program can parse the course information from Albert for the future use

Pre-conditions: Working Albert

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected**  **Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1 | Enter “make parse” to the terminal | parsing.py | File “data.html” should be generated if parsing is successful. | File “data.html” is created with the course information. | Pass |  |

**Postconditions:** the html file with the course information is generated

Test Case 2

Project Name: Automated Course Scheduling System

Test Case ID: 0102 Test Designed by: Akhat Suleimenov

Test Priority (Low/Medium/High): H Test Designed date: 26.11.21

Module Name: Import Course Information Test Executed by: Dariga Shokayeva

Test Title: Parsing data from malfunctional Albert Test Execution date: 26.11.21

Description: the python script for parsing data is run to get course information from not functioning Albert

Objective: test if the program can parse the course information from Albert for the future use

Pre-conditions: Working Albert

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected**  **Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1 | Enter “make parse” to the terminal | parsing.py | Error message is displayed. Data from backup file is written to “data.html” file | Error message is shown. File “data.html” is created with the course information. | Pass |  |

**Postconditions:** the html file with the course information from the backup file is generated

Test Case 3

Project Name: Automated Course Scheduling System

Test Case ID: 0103 Test Designed by: Akhat Suleimenov

Test Priority (Low/Medium/High): L Test Designed date: 26.11.21

Module Name: Import Course Information Test Executed by: Dariga Shokayeva

Test Title: Parsing data with no Internet connection Test Execution date: 26.11.21

Description: the python script for parsing data is run to get course information from Albert while there is no active Internet connection

Objective: test if the program can parse the course information from Albert for the future use

Pre-conditions: Working Albert

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected**  **Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1 | Enter “make parse” to the terminal | parsing.py | Error message is displayed. Data from backup file is written to “data.html” file | Error message is shown. File “data.html” is created with the course information. | Pass |  |

**Postconditions:** the html file with the course information from the backup file is generated

Test Case 4

Project Name: Automated Course Scheduling System

Test Case ID: 0201 Test Designed by: Madina Yeleukina

Test Priority (Low/Medium/High): H Test Designed date: 27.11.21

Module Name: User Authorization Test Executed by: Dariga Shokayeva

Test Title: User authorization attempt with non-NYU account Test Execution date: 28.11.21

Description: user attempts to authorize with non-NYU Gmail account

Objective: test if the Google authorization denies access for non-NYU users

Pre-conditions: None

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected**  **Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1 | User attempts to authorize to ACSS with the Gmail account | [shokayeva@gmail.com](mailto:shokayeva@gmail.com) | Authorization error “This client is restricted to users within its organization.” | Authorization error “This client is restricted to users within its organization.” | Pass |  |

**Postconditions:** User is not authorized to the system

Test Case 5

Project Name: Automated Course Scheduling System

Test Case ID: 0202 Test Designed by: Madina Yeleukina

Test Priority (Low/Medium/High): H Test Designed date: 27.11.21

Module Name: User Authorization Test Executed by: Dariga Shokayeva

Test Title: User authorization attempt with NYU affiliated Gmail account

Test Execution date: 28.11.21

Description: user attempts to authorize with NYU affiliated Gmail account

Objective: test if the Google authorization gives access for NYU users

Pre-conditions: None

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected**  **Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1 | User attempts to authorize to ACSS with the Gmail account | ds6058@nyu.edu | User is signed in | User is signed in | Pass |  |

**Postconditions:** Transfer to main page, courses are shown

Test Case 6

Project Name: Automated Course Scheduling System

Test Case ID: 0203 Test Designed by: Madina Yeleukina

Test Priority (Low/Medium/High): L Test Designed date: 27.11.21

Module Name: User Authorization Test Executed by: Dariga Shokayeva

Test Title: User authorization attempt to ACSS from the unknown server

Test Execution date: 28.11.21

Description: user visits the ACSS from the server unauthorized for Google API services

Objective: test if the ACSS runs only on the servers added to Google API services by us

Pre-conditions: None

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected**  **Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1 | User attempts to authorize to ACSS from the unknown for Google API services | Localhost:7800 | Authorization error raised by Google services | Localhost does not work, error | Fail | Fixed by adding localhost address into the gmail database |

**Postconditions:** User is not authorized to the system

Test Case 7

Project Name: Automated Course Scheduling System

Test Case ID: 0301 Test Designed by: Dariga Shokayeva

Test Priority (Low/Medium/High): M Test Designed date: 28.11.21

Module Name: Search/Add Classes Test Executed by: Shyngys Karishev

Test Title: User Adds a Class Test Execution date: 1.12.21

Description: user clicks ‘Add’ button to select a course

Objective: test if the selected course appears in the “Courses Picked”

Pre-conditions: course information import, user authorization

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected**  **Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1 | User presses “Add” button | Add “ACS-UH 1010X - Anthropology and the Arab World” | ACS-UH 1010X - Course code appears in the “Courses Picked” and the label for that course changes to “Added” | ACS-UH 1010X - Course code appears in the “Courses picked” and the label for that course changes to “Added” | Pass |  |

**Postconditions:** one course is stored in the “Courses Picked”

Test Case 8

Project Name: Automated Course Scheduling System

Test Case ID: 0302 Test Designed by: Dariga Shokayeva

Test Priority (Low/Medium/High): M Test Designed date: 28.11.21

Module Name: Search/Add Classes Test Executed by: Shyngys Karishev

Test Title: User Removes a Picked Class | Course List Test Execution date: 1.12.21

Description: user clicks ‘Added’ button after selecting this course before

Objective: test if the selected course disappears in the “Courses Picked” and marked as not yet added in the course list

Pre-conditions: course information import, user authorization

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected**  **Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1 | User presses “Added” button for a picked class | Added “ACS-UH 1010X - Anthropology and the Arab World” | ACS-UH 1010X - Course code disappears in the “Courses Picked” and the label for that course changes back to “Add” | ACS-UH 1010X - Course code disappears in the “Courses picked” and the label for that course changes back to “Add” | Pass |  |

**Postconditions:** the class is removed from the “Courses Picked”

Test Case 9

Project Name: Automated Course Scheduling System

Test Case ID: 0303 Test Designed by: Dariga Shokayeva

Test Priority (Low/Medium/High): M Test Designed date: 28.11.21

Module Name: Search/Add Classes Test Executed by: Shyngys Karishev

Test Title: User Removes a Class | Courses Picked List Test Execution date: 1.12.21

Description: user clicks remove button for the class in “Course Picked” after selecting this course before

Objective: test if the selected course disappears in the “Courses Picked” and is marked us not yet added in the course list

Pre-conditions: course information import, user authorization

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected**  **Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1 | User presses remove button for a picked class in the “Courses picked” | Added “ACS-UH 1010X - Anthropology and the Arab World” | ACS-UH 1010X - Course code disappears in the “Courses Picked” and the label for that course changes back to “Add” | ACS-UH 1010X - Course code disappears in the “Courses picked” and the label for that course changes back to “Add” | Pass |  |

**Postconditions:** the class is removed from the “Courses Picked”

Test Case 10

Project Name: Automated Course Scheduling System

Test Case ID: 0304 Test Designed by: Dariga Shokayeva

Test Priority (Low/Medium/High): L Test Designed date: 28.11.21

Module Name: Search/Add Classes Test Executed by: Shyngys Karishev

Test Title: Search Classes by Partial Course Code Test Execution date: 1.12.21

Description: user enters the part of course code to search for courses

Objective: test if the courses with the entered by user code are found and displayed

Pre-conditions: course information import, user authorization

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected**  **Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1 | User enters the part of course code | “ACS” | Courses with ACS code or courses with ACS in the course name should be listed. | Courses with ACS code are listed. | Pass |  |

**Postconditions:** the list of courses with the code entered by user appears

Test Case 11

Project Name: Automated Course Scheduling System

Test Case ID: 0305 Test Designed by: Dariga Shokayeva

Test Priority (Low/Medium/High): L Test Designed date: 28.11.21

Module Name: Search/Add Classes Test Executed by: Shyngys Karishev

Test Title: Search Class by Course Code Test Execution date: 1.12.21

Description: user enters the course code to search for a course

Objective: test if the course with the entered by user code is found and displayed

Pre-conditions: course information import, user authorization

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected**  **Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1 | User enters the course code | “CS-UH 1050” | Unique course with all its sections should be found and displayed | All Data Structures sections are listed, no other course listings appear | Pass |  |

**Postconditions:** the list of sections of the course associated with the code entered by user appears

Test Case 12

Project Name: Automated Course Scheduling System

Test Case ID: 0306 Test Designed by: Dariga Shokayeva

Test Priority (Low/Medium/High): M Test Designed date: 28.11.21

Module Name: Search/Add Classes Test Executed by: Shyngys Karishev

Test Title: Lowercase Search Input Test Execution date: 1.12.21

Description: user inputs the string that is fully lowercase to search for a course

Objective: test if the search is not case-sensitive

Pre-conditions: course information import, user authorization

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected**  **Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1 | User enters the input for search in lowercase letters | “introduction for computer science” | Introduction to Computer Science sections should be found and displayed | Introduction to Computer Science sections should be found and displayed | Pass |  |

**Postconditions:** the list of sections of the course associated with the user input appears

Test Case 13

Project Name: Automated Course Scheduling System

Test Case ID: 0307 Test Designed by: Dariga Shokayeva

Test Priority (Low/Medium/High): L Test Designed date: 28.11.21

Module Name: Search/Add Classes Test Executed by: Shyngys Karishev

Test Title: Uppercase Search Input Test Execution date: 1.12.21

Description: user inputs the string that is fully uppercase to search for a course

Objective: test if the search is not case-sensitive

Pre-conditions: course information import, user authorization

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected**  **Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1 | User enters the input for search in uppercase letters | “INTRODUCTION FOR COMPUTER SCIENCE” | Introduction to Computer Science sections should be found and displayed | Introduction to Computer Science sections should be found and displayed | Pass |  |

**Postconditions:** the list of sections of the course associated with the user input appears

Test Case 14

Project Name: Automated Course Scheduling System

Test Case ID: 0308 Test Designed by: Dariga Shokayeva

Test Priority (Low/Medium/High): L Test Designed date: 28.11.21

Module Name: Search/Add Classes Test Executed by: Shyngys Karishev

Test Title: User Removes a Class after Search | Course List Test Execution date: 1.12.21

Description: user clicks ‘Added’ button after searching and adding that class

Objective: test if the selected course disappears in the “Courses Picked” and is marked us not yet added in the course list

Pre-conditions: course information import, user authorization

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected**  **Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1 | User presses “Added” button for a picked class after search | Added “ACS-UH 1010X - Anthropology and the Arab World” | ACS-UH 1010X - Course code disappears in the “Courses Picked” and the label for that course changes back to “Add” | ACS-UH 1010X - Course code disappears in the “Courses picked” and the label for that course changes back to “Add” | Pass |  |

**Postconditions:** the class is removed from the “Courses Picked”

Test Case 15

Project Name: Automated Course Scheduling System

Test Case ID: 0309 Test Designed by: Dariga Shokayeva

Test Priority (Low/Medium/High): M Test Designed date: 28.11.21

Module Name: Search/Add Classes Test Executed by: Shyngys Karishev

Test Title: User Removes a Class after Search | Courses Picked List

Test Execution date: 1.12.21

Description: user searches for a class then clicks remove button for the class in “Course Picked”

Objective: test if the selected course disappears in the “Courses Picked” and is marked us not yet added in the course list

Pre-conditions: course information import, user authorization

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected**  **Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1 | User presses remove button for a class in the “Courses picked” after search | Added “ACS-UH 1010X - Anthropology and the Arab World” | ACS-UH 1010X - Course code disappears in the “Courses Picked” and the label for that course changes back to “Add” | ACS-UH 1010X - Course code disappears in the “Courses picked” and the label for that course changes back to “Add” | Pass |  |

**Postconditions:** the class is removed from the “Courses Picked”

Test Case 16

Project Name: Automated Course Scheduling System

Test Case ID: 0310 Test Designed by: Dariga Shokayeva

Test Priority (Low/Medium/High): L Test Designed date: 28.11.21

Module Name: Search/Add Classes Test Executed by: Shyngys Karishev

Test Title: Maximum number of classes exceeded Test Execution date: 1.12.21

Description: user selects more than 9 classes where 9 is the maximum permitted number of classes selected

Objective: test if the error message is displayed if user selects more than permitted number of classes

Pre-conditions: course information import, user authorization

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected**  **Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1 | User tries to add 10th class to the schedule |  | Error message “maximum of 9 courses allowed” is displayed | Error message “maximum of 9 courses allowed” is displayed | Pass |  |

**Postconditions:** the nine courses selected by user previously is stored

Test Case 17

Project Name: Automated Course Scheduling System

Test Case ID: 0311 Test Designed by: Dariga Shokayeva

Test Priority (Low/Medium/High): H Test Designed date: 28.11.21

Module Name: Search/Add Classes Test Executed by: Shyngys Karishev

Test Title: Expected number of courses selected Test Execution date: 1.12.21

Description: user selects courses with the total number of selected courses between 1 and 9 (both inclusive) and hits submit

Objective: test if ACSS recognizes valid input and creates schedules

Pre-conditions: course information import, user authorization

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected**  **Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1 | User selects the classes and presses ‘Submit’ button | 5 courses | Upon hitting ‘Submit’  button, schedules should be created and displayed | Upon hitting ‘submit’ button, schedules are created and displayed | Pass |  |

**Postconditions:** website generates all the possible schedules and displays them in a calendar format under the search block

Test Case 18

Project Name: Automated Course Scheduling System

Test Case ID: 0312 Test Designed by: Dariga Shokayeva

Test Priority (Low/Medium/High): L Test Designed date: 28.11.21

Module Name: Search/Add Classes Test Executed by: Shyngys Karishev

Test Title: Expected number of courses selected Test Execution date: 1.12.21

Description: user selects 9 (boundary count) courses and hits submit

Objective: test if ACSS recognizes valid input and creates schedules

Pre-conditions: course information import, user authorization

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected**  **Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1 | User selects the classes and presses ‘Submit’ button | 9 courses | Upon hitting ‘Submit’  button, schedules should be created and displayed | Program crashed; last course was overwritten by some dummy value | Fail | Fixed by modifying the loop iterations to make sure last one executes |

**Postconditions:** website generates all the possible schedules and displays them in a calendar format under the search block

Test Case 19

Project Name: Automated Course Scheduling System

Test Case ID: 0313 Test Designed by: Dariga Shokayeva

Test Priority (Low/Medium/High): L Test Designed date: 28.11.21

Module Name: Search/Add Classes Test Executed by: Shyngys Karishev

Test Title: Expected number of courses selected Test Execution date: 1.12.21

Description: user selects 1 (boundary count) courses and hits submit

Objective: test if ACSS recognizes valid input and creates schedules

Pre-conditions: course information import, user authorization

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected**  **Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1 | User selects the classes and presses ‘Submit’ button | 1 course | Upon hitting ‘Submit’  button, schedules should be created and displayed | Upon hitting ‘submit’ button, schedules are created and displayed | Pass |  |

**Postconditions:** website generates all the possible schedules and displays them in a calendar format under the search block

Test Case 20

Project Name: Automated Course Scheduling System

Test Case ID: 0314 Test Designed by: Dariga Shokayeva

Test Priority (Low/Medium/High): M Test Designed date: 28.11.21

Module Name: Search/Add Classes Test Executed by: Shyngys Karishev

Test Title: No classes selected Test Execution date: 1.12.21

Description: user does not select any class and hits submit

Objective: test if ACSS recognizes valid input and creates schedules

Pre-conditions: course information import, user authorization

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected**  **Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1 | User selects no classes and presses ‘Submit’ button |  | Upon hitting ‘Submit’  Button, error message is displayed | No error message appeared. Program crashed | Fail | Fixed by displaying error message and not going further after submit button |

**Postconditions:** Error message indicating that the user should pick at least 1 course

Test Case 21

Project Name: Automated Course Scheduling System

Test Case ID: 0401 Test Designed by: Shyngys Karishev

Test Priority (Low/Medium/High): H Test Designed date: 1.12.21

Module Name: Create schedule Test Executed by: Akhat Suleimenov

Test Title: Create schedule with valid inputs Test Execution date: 1.12.21

Description: Run schedule script with all the valid files

Objective: Check if valid schedule is being created

Pre-conditions: Courses picked by the user

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected**  **Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1 | Run “make calendar\_json” | create\_json\_calendar.py | Json file containing the list of all possible schedules with given course | Json file will all the possible schedules | Pass |  |

**Postconditions:** calendar.json file created

Test Case 22

Project Name: Automated Course Scheduling System

Test Case ID: 0402 Test Designed by: Shyngys Karishev

Test Priority (Low/Medium/High): H Test Designed date: 1.12.21

Module Name: Create schedule Test Executed by: Akhat Suleimenov

Test Title: Create schedule without file where all classes information is saved

Test Execution date: 1.12.21

Description: Run schedule script without courses\_dict.json

Objective: Check how program behaves when json file is missing

Pre-conditions: Courses picked by the user

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected**  **Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1 | Run “make calendar\_json” | create\_json\_calendar.py | Error message stating that the file is missing | Error message stating that the file is missing | Pass |  |

**Postconditions:** Error message indicating that a file is missing

Test Case 23

Project Name: Automated Course Scheduling System

Test Case ID: 0403 Test Designed by: Shyngys Karishev

Test Priority (Low/Medium/High): H Test Designed date: 1.12.21

Module Name: Create schedule Test Executed by: Akhat Suleimenov

Test Title: Create schedule without file with list of selected classes

Test Execution date: 1.12.21

Description: Run schedule script without selected.json

Objective: Check how program behaves when json file is missing

Pre-conditions: Courses picked by the user

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected**  **Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1 | Run “make calendar\_json” | create\_json\_calendar.py | Error message stating that the selected.json is missing | Error message stating that the selected.json is missing | Pass |  |

**Postconditions:** Error message indicating that a selected.json is missing

Test Case 24

Project Name: Automated Course Scheduling System

Test Case ID: 0404 Test Designed by: Shyngys Karishev

Test Priority (Low/Medium/High): L Test Designed date: 1.12.21

Module Name: Create schedule Test Executed by: Akhat Suleimenov

Test Title: Create schedule without both required courses Test Execution date: 1.12.21

Description: Run schedule script without selected.json and courses\_dict.json

Objective: Check how program behaves when json files a missing

Pre-conditions: Courses picked by the user

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected**  **Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1 | Run “make calendar\_json” | create\_json\_calendar.py | Error message stating that the selected.json and courses\_dict.json is missing | Error message stating that the selected.json and courses\_dict.json are missing | Pass |  |

**Postconditions:** Error message indicating that selected.json and courses\_dict.json are missing

Test Case 25

Project Name: Automated Course Scheduling System

Test Case ID: 0405 Test Designed by: Shyngys Karishev

Test Priority (Low/Medium/High): L Test Designed date: 1.12.21

Module Name: Create schedule Test Executed by: Akhat Suleimenov

Test Title: Create schedule with empty list of selected courses Test Execution date: 1.12.21

Description: Run schedule script with empty selected.json

Objective: Check how program behaves when there are no selected courses

Pre-conditions: Courses picked by the user

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected**  **Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1 | Run “make calendar\_json” | create\_json\_calendar.py | Error message stating that the list is empty | Program crashed with JSONDecoderError | Fail | Fixed by adding try except block in the reading file function |

**Postconditions:** Error message indicating that selected.json is empty

Test Case 26

Project Name: Automated Course Scheduling System

Test Case ID: 0406 Test Designed by: Shyngys Karishev

Test Priority (Low/Medium/High): L Test Designed date: 1.12.21

Module Name: Create schedule Test Executed by: Akhat Suleimenov

Test Title: Create schedule with empty list of courses dictionary

Test Execution date: 1.12.21

Description: Run schedule script with empty courses\_dict.json

Objective: Check how program behaves when there are no courses

Pre-conditions: Courses picked by the user

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected**  **Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1 | Run “make calendar\_json” | create\_json\_calendar.py | Error message stating that the courses file is empty | Program crashed with JSONDecoderError | Fail | Fixed by adding try except block in the reading file function |

**Postconditions:** Error message indicating that courses\_dict.json is empty

Test Case 27

Project Name: Automated Course Scheduling System

Test Case ID: 0407 Test Designed by: Shyngys Karishev

Test Priority (Low/Medium/High): L Test Designed date: 1.12.21

Module Name: Create schedule Test Executed by: Akhat Suleimenov

Test Title: Create schedule with list of schedules of wrong format

Test Execution date: 1.12.21

Description: Run schedule script with incorrect information inside of selected.json

Objective: Check how program behaves when the format is not the one expected

Pre-conditions: Courses picked by the user

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected**  **Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1 | Run “make calendar\_json” | create\_json\_calendar.py | Error message stating that something is wrong with the file format | Program crashed with ValueError | Fail | Fixed by adding try except block in the correct\_selected\_schedule function |

**Postconditions:** Error message indicating that selected.json is wrongly formatted

Test Case 28

Project Name: Automated Course Scheduling System

Test Case ID: 0408 Test Designed by: Shyngys Karishev

Test Priority (Low/Medium/High): L Test Designed date: 1.12.21

Module Name: Create schedule Test Executed by: Akhat Suleimenov

Test Title: Create schedule where fixed courses have time conflict

Test Execution date: 1.12.21

Description: Run schedule script two fixed courses that have time conflict

Objective: Check how program behaves when the given fixed courses clash

Pre-conditions: Courses picked by the user

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected**  **Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1 | Run “make calendar\_json” | create\_json\_calendar.py | Calendar.json file created with empty list inside | Calendar.json created with empty list | Pass |  |

**Postconditions:** Calendar.json file with empty list

Test Case 29

Project Name: Automated Course Scheduling System

Test Case ID: 0409 Test Designed by: Shyngys Karishev

Test Priority (Low/Medium/High): M Test Designed date: 1.12.21

Module Name: Create schedule Test Executed by: Akhat Suleimenov

Test Title: Create schedule where one of the ids passed does not exist

Test Execution date: 1.12.21

Description: Run schedule script with a wrong id in the selected.json file

Objective: Check how program behaves when one of the ids is not present in the dictionary

Pre-conditions: Courses picked by the user

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected**  **Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1 | Run “make calendar\_json” | create\_json\_calendar.py | Error message stating that the following id does not exist | Error message stating that the following id does not exist | Pass |  |

**Postconditions:** Error message stating that the following id does not exist

Test Case 30

Project Name: Automated Course Scheduling System

Test Case ID: 0410 Test Designed by: Shyngys Karishev

Test Priority (Low/Medium/High): L Test Designed date: 1.12.21

Module Name: Create schedule Test Executed by: Akhat Suleimenov

Test Title: Create schedule where one of course has a wrong format in the class structure

Test Execution date: 1.12.21

Description: Run program by modifying the ‘days’ and ‘start-date’ keys of a class to ‘day’ and ‘start-dat’

Objective: Check how program behaves when wrong key is trying to be accessed

Pre-conditions: Courses picked by the user

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected**  **Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1 | Run “make calendar\_json” | create\_json\_calendar.py | Error message stating that the course is malfunctional. Calendar created | Wrong course name was being blamed for the error | Fail | Fixed by adding two try except blocs for both fixed and unfixed courses |

**Postconditions:** Error message stating which course has error and calendar created if possible.

Test Case 31

Project Name: Automated Course Scheduling System

Test Case ID: 0411 Test Designed by: Shyngys Karishev

Test Priority (Low/Medium/High): H Test Designed date: 1.12.21

Module Name: Create schedule Test Executed by: Akhat Suleimenov

Test Title: Run the whole script at once Test Execution date: 03.12.21

Description: Run the make all

Objective: Check if calendar modules are integrated properly

Pre-conditions: User selected classes json file exists

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected**  **Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1 | Run “make all” script | constants.py course\_class.py parsing.py html\_to\_json.py create\_json\_calendar.py | Data is parsed from Albert. Calendar of all possible schedules from given classes is created | Data is parsed. calendar.json file created along other json files | Pass |  |

**Postconditions:** calendar.json file created

Test Case 32

Project Name: Automated Course Scheduling System

Test Case ID: 0412 Test Designed by: Shyngys Karishev

Test Priority (Low/Medium/High): H Test Designed date: 1.12.21

Module Name: Create schedule Test Executed by: Akhat Suleimenov

Test Title: Run the whole script at once but without selected.json file

Test Execution date: 03.12.21

Description: Run the make all without having selected.json file

Objective: Check if calendar modules are integrated properly

Pre-conditions: User selected classes json file exists

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected**  **Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1 | Run “make all” script | constants.py course\_class.py parsing.py html\_to\_json.py create\_json\_calendar.py | Data parsed from Albert. Error message stating selected.json file is missing printed | Data parsed from Albert. Error message stating selected.json file is missing printed | Pass |  |

**Postconditions:** Data parsed into data.html file. Error message displayed

Test Case 33

Project Name: Automated Course Scheduling System

Test Case ID: 0501 Test Designed by: Dariga Shokayeva

Test Priority (Low/Medium/High): M Test Designed date: 30.12.21

Module Name: Displaying Schedules Test Executed by: Madina Yeleukina

Test Title: Empty schedules list Test Execution date: 01.12.21

Description: The system receives empty schedules list

Objective: Test the system’s response if it receives empty schedules list

Pre-conditions: Generated schedules list

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected**  **Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1 | Read calendar.json file | Empty file | Message appears stating that schedules cannot be generated based on user’s picks | Nothing appears on the screen | Fail | Fixed by checking received schedules list for emptiness |

**Postconditions:** Displayed schedules

Test Case 34

Project Name: Automated Course Scheduling System

Test Case ID: 0502 Test Designed by: Dariga Shokayeva

Test Priority (Low/Medium/High): M Test Designed date: 30.11.2021

Module Name: Displaying Schedules Test Executed by: Madina Yeleukina

Test Title: Valid input Test Execution date: 01.12.21

Description: The system receives a valid schedules list

Objective: Test the basic functionality of the system with a valid schedules list

Pre-conditions: Generated schedules list

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected**  **Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1 | Read calendar.json file | 5 generated schedules | 5 schedules created: 4 hid and 1 displayed | 5 schedules created: 4 hid and 1 displayed | Pass |  |

**Postconditions:** Displayed schedules

Test Case 35

Project Name: Automated Course Scheduling System

Test Case ID: 0503 Test Designed by: Dariga Shokayeva

Test Priority (Low/Medium/High): H Test Designed date: 30.11.21

Module Name: Displaying Schedules Test Executed by: Madina Yeleukina

Test Title: Excessive schedules list Test Execution date: 01.12.21

Description: The system receives a schedules list containing an excessive number of schedules

Objective: Test the system’s response if it receives a schedules list containing an excessive number of schedules generated

Pre-conditions: Generated schedules list

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected**  **Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1 | Read calendar.json file | 100 schedules | Message appears stating that there are too many possible schedules and only some of them displayed | System loads too long | Fail | Fixed by checking received schedules list for excess |

**Postconditions:** Displayed schedules

Test Case 36

Project Name: Automated Course Scheduling System

Test Case ID: 0504 Test Designed by: Dariga Shokayeva

Test Priority (Low/Medium/High): M Test Designed date: 30.11.21

Module Name: Displaying Schedules Test Executed by: Madina Yeleukina

Test Title: First schedule displayed Test Execution date: 01.12.21

Description: The user wants to see the previous schedule when the system is displaying the first schedule

Objective: Test the system’s response if the user wants to see the previous schedule of the first generated schedule

Pre-conditions: Generated schedules list

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected**  **Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1 | Read calendar.json file | 5 generated schedules | 5 schedules created: 4 hid and 1 displayed | 5 schedules created: 4 hid and 1 displayed | Pass |  |
| 2 | Press previous schedule button |  | The last schedule displayed | The first schedule displayed | Fail | Fixed by changing previous button’s algorithm |

**Postconditions:** Displayed schedules

Test Case 37

Project Name: Automated Course Scheduling System

Test Case ID: 0505 Test Designed by: Dariga Shokayeva

Test Priority (Low/Medium/High): M Test Designed date: 30.11.21

Module Name: Displaying Schedules Test Executed by: Madina Yeleukina

Test Title: Last schedule displayed Test Execution date: 01.12.21

Description: The user wants to see the next schedule when the system is displaying the last schedule

Objective: Test the system’s response if the user wants to see the next schedule of the last generated schedule

Pre-conditions: Generated schedules list

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected**  **Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1 | Read calendar.json file | 5 generated schedules | 5 schedules created: 4 hid and 1 displayed | 5 schedules created: 4 hid and 1 displayed | Pass |  |
| 2 | Press next schedule button |  | The first schedule displayed | The last schedule displayed | Fail | Fixed by changing next button’s algorithm |

**Postconditions:** Displayed schedules

Test Case 38

Project Name: Automated Course Scheduling System

Test Case ID: 0506 Test Designed by: Dariga Shokayeva

Test Priority (Low/Medium/High): H Test Designed date: 01.12.21

Module Name: Displaying Schedules Test Executed by: Madina Yeleukina

Test Title: Absence of schedules list Test Execution date: 01.12.21

Description: The system does not receive any generated schedules list

Objective: Test the system’s response if the calendar.json file is not present

Pre-conditions: Generated schedules list

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected**  **Result** | **Actual Result** | **Status**  **(Pass/Fail)** | **Notes** |
| 1 | Read calendar.json file |  | Message appears stating that schedules cannot be generated | System crashes | Fail | Fixed by checking the presence of calendar.json file |

**Postconditions:** Displayed schedules