Testing of Expense Tracker Application for T1A3 - By Ishan Acharya. Tests conducted on 16 December 2023

Test #	Test Case	Test Steps	Expected Result	Actual Result	Status	Issue Identified	Status after issue addressed
1	#1 - To test that the record_income function is working as intended by adding an income entry with income name 'weekly salary', income amount of '$5,000', and income category 'Salary', with the datetime package adding today's current date (16 December 2023). This entry should be displayed in a table format with the tabulate package.	1. Run the application 2. Enter input 1 to select 'create new expense tracker' from the main menu 3. Enter input 1 to select 'Add income entry' from the sub menu 4. Enter income name 'weekly salary' when prompted 5. Enter income amount '$5,000' when prompted 6. Enter input 1 to select 'salary' income category	1. User should see a confirmation message displayed that their income entry has been recorded successfully at completion of step 6. 2. Income should be printed at the end of step 6 showing the income entry with the entry name (weekly salary), entry amount ($5,000), and category (salary).	1. A confirmation message is displayed stating that the income entry has been successfully added. 2. The income entry is printed in a table whilst displaying the name (weekly salary), amount ($5,000), and today's date (2023/12/16), however is NOT displaying the income category.	The record_income function works as intended, however the tabulated data does not display income category	Identified issue in the code within record_income function to print tabulated data was missing new_income.category, which was causing the data to be printed without the income category included. This issue has been rectified with the inclusion of new_income.category when printing the tabulated data.	The record_income function works as intended
1	#2 - To test that the record_expense function is working as intended by adding an expense entry with expense name 'weekly rent', expense amount of '$800', and expense category 'Rent', with the datetime package adding today's current date (16 December 2023). This entry should be displayed in a table format with the tabulate package.	1. Run the application 2. Enter input 1 to select 'create new expense tracker' from the main menu 3. Enter input 2 to select 'Add expense entry' from the sub menu 4. Enter expense name 'weekly rent' when prompted 5. Enter expense amount '$800' when prompted 6. Enter input 3 to select 'salary' expense category	1. User should see a confirmation message displayed that their expense entry has been recorded successfully at completion of step 6. 2. Expense should be printed at the end of step 6 showing the expense entry with the expense name (weekly rent), expense amount ($800), and category (rent).	1. A confirmation message is displayed stating that the expense entry has been successfully added. 2. The expense entry is printed in a table whilst displaying the name (weekly rent), amount ($800) and today's date (2023/12/16), however is NOT displaying the expense category.	The record_expense function works as intended, however the tabulated data does not display expense category	Identified issue in the code within record_expense function to print tabulated data was missing new_expense.category, which was causing the data to be printed without the expense category included. This issue has been rectified with the inclusion of new_expense.category when printing the tabulated data.	The record_expense function works as intended
2	#1 - To test that the save_data_to_file function is working as intended by saving expense tracker file from test #1, with the income and expense data present, by inputting the file name 'my_data', and saving it to the user's desktop location on their local machine. The saved data file should contain the recorded income and expense data from test #1 on it.	(continued from test#1 step 6) 1. Enter input 4 to select 'save budget data' from the sub menu 2. Enter file name 'my_data' when prompted to enter file name to save budget data 3. Open saved file 'my_data' in user's desktop location	1. User should be able to input 'my_data' to provide the file name for the budget data that is to be saved. 2. User should see a confirmation message outlining that the budget data has been saved successfully to the user's desktop location, with the file path outlined, on their local machine. 3. The saved data with the file name 'my_data' should be present in the user's desktop location on their local machine. 4. After opening the 'my_data' file, it should be a .txt file containing the income entry data and expense entry data from test #1, which includes the income/expense name, income/expense amount, income/expense category, and income/expense date.	1. User can enter file name 'my_data' to save the budget data. 2. A confirmation message is displayed stating that the budget data has been successfully saved to the user's desktop location on their local machine, with the file path outlined. 3. The saved data 'my_data' is present on the user's desktop location on their local machine. 4. The 'my_data'.txt file contains both the income and expense entries from test #1 and includes the income/expense names, income/expense amounts, income/expense categories, and the income/expense dates.	The save_data_to_file function works as intended	N/A	N/A
2	#2 - To test that the load_expense_tracker function is working as intended by loading the saved data file from test #2, containing the data from test case #1, by inputting the file name 'my_data' from the user's desktop. Once the file has been loaded, it should restore the income and expense data from test #1 in the view_budget function.	1. Run the application 2. Enter input 2 to select 'load existing expense tracker' from the main menu 3. Enter file name 'my_data' when prompted to enter file name of the saved data to load 4. Enter input 3 to select 'view your budget for this month' from the sub menu 5. Enter input 1 to select 'December 2023' from the list of available months 6. Enter input 1 to select 'View your entries for a specific month' from the list of view options	1. User should be able to input 'my_data' to provide the file name to identify the file for the budget data file that is to be loaded. 2. User should see a confirmation message outlining that the saved budget data 'my_data' has been successfully loaded from the user's desktop location, with the file path outlined, on their local machine. 3. The loaded data from the 'my_data' file should display the income and expense entries that were recorded in test #1 within the view_budget function.	1. User can enter file name 'my_data' to identify the budget data file to load from their desktop location. 2. A confirmation message is displayed stating that the data has been successfully loaded from the user's desktop location, with the file path outlined to their desktop location, from the user's local machine. 3. Once the file has been loaded, the income and expense entries from test #1 have been restored and can be viewed when going into the 'view your entries for a specific month' option, which is within the function of the view_budget function.	The load_expense_tracker function works as intended	N/A	N/A