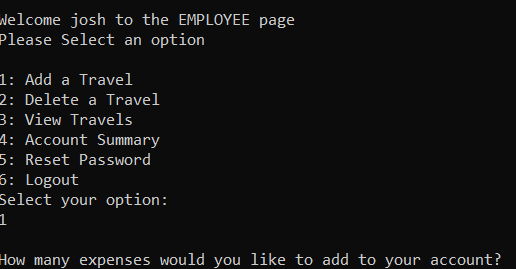
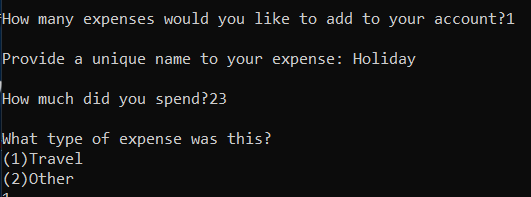
This document evaluates the performance of the attached program.

This is how each input requirement was met:

* **Number of journeys completed**
* **What the claim is for**
* **Cost of claim**



This step requires creating an employee account and logging into the system where travels can be added, and the various inputs can be entered for the system to process.

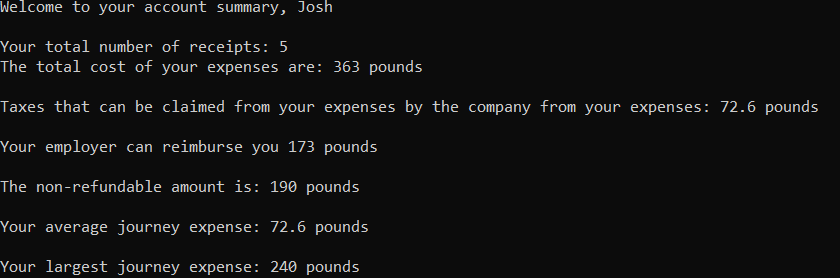


This is how each output requirement was met:

Receipt details

* **Total cost of the expenses combined**
* **Tax that can reclaimed by the company, calculated at 20% of the total cost**
* **Amount that will need to be paid to the employee**
* **They will be awarded 100% of the travel cost**
* **Employees can only claim up to £50 of expenses, anything over will need to be covered by the employee**
* **The non-refundable amount**
* **The average payment required**
* **The largest payment made**

Under the employee page, there is an account summary option which adds the required outputs under one screen



**Evaluation**

The basic inputs and outputs of the assignment was met, wrapped around a menu log in system that separates the employees and admin accounts, both of which holds different controls to the data. Thought due to this, an admin will require a separate employee account to add tasks as this is not part of their menu. However due to the high modularity design of the system, additional features can be implemented easily by recycling existing functions.

Though functioning for the inputs and outputs required, the menu system itself is incomplete only having the login system completed and the admin feature is yet left to desire more functioning features. Improvements can include having admins have the same features under their page as the employees along with their additional features that can control the data of employees, but not other admins.

The employee page is fairly well developed, only lacking the recover account feature and a delete journey feature, as they are able to access all their expenses and summaries under a few clicks. The addition of more expenses could be designed to be a bit clearer for users to use and an issue arises that if you accidentally press too many journeys to add, e.g. 100, you will be stuck to add all 100 journeys. A feature to confirm the number of expenses to enter or be released from the loop at every few steps would be useful to counter this. Confirmation can also be added to other sensitive data such as passwords.

However, a positive for the system is the use of saving data externally in the form of CSVs. This allows a more permanent storage of data, but passwords are currently seen directly if a user has access to the Accountfile.csv file. An encryption would be a good way to counter this, even if it’s a simple shift of the words by a few letters.