Terminal Application

TRAVEL LOG

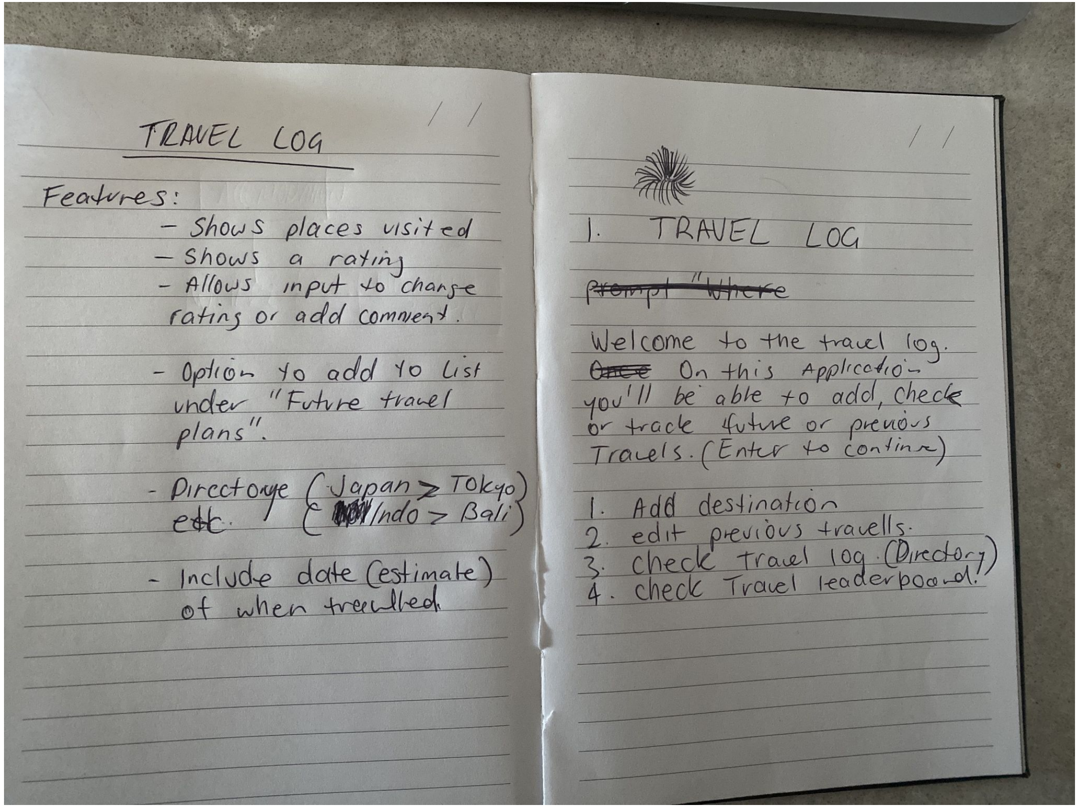
Term 1, Coder Academy

By Jye Calder-Mason

DESIGN PROCESS

Straight from the start I knew I was going to make the travel log, I had the idea as I was reading the assignment. It's much easier to work on something that you enjoy because you want to see a good outcome, so working with this was a breeze. Travelling was a big part of my life pre COVID, so going back and referring to old holidays was a fun time.

This first photo was a bit of scribble I wrote down while being assigned the assignment. Was very helpful in the beginning as I really didn't know what I could and couldn't do within terminal.



/ /

TRAVEL LOG

Features:

- Shows places visited
- Shows a rating
- Allows input to change rating or add comment.

- Option to add to list under "Future travel plans".

- Directory (Japan > Tokyo)
  etc. (Indo > Bali)

- Include date (estimate) of when travelled.

/ /

1. TRAVEL LOG

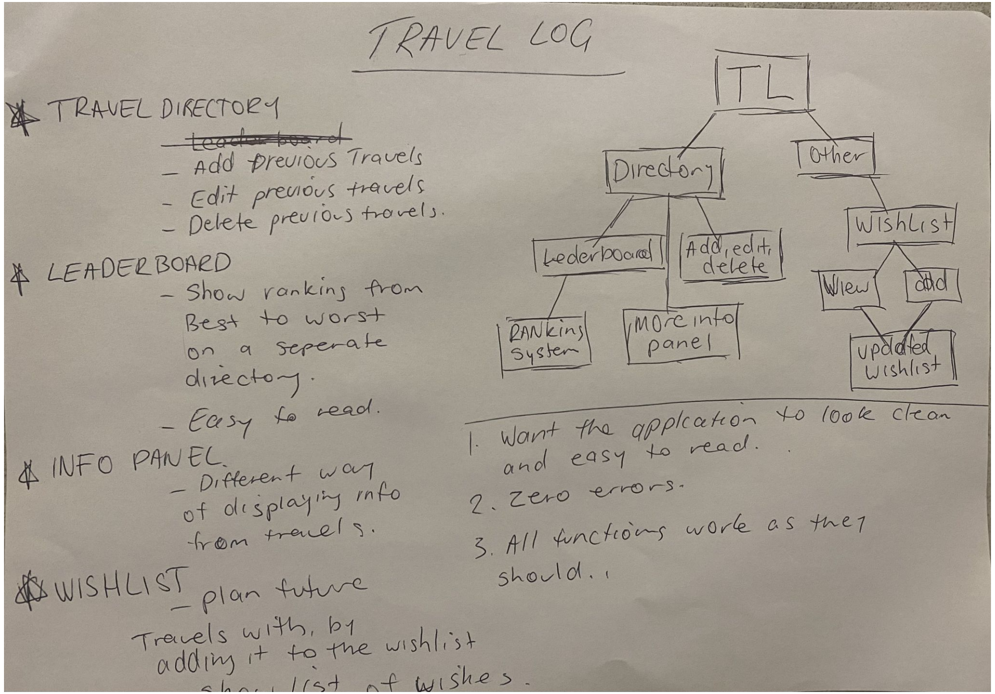
~~Prompt "Where~~

Welcome to the travel log. ~~Once~~ On this Application you'll be able to add, check or track future or previous Travels. (Enter to continue)

1. Add destination
2. edit previous travells.
3. check Travel log. (Directory)
4. check Travel leaderboard!

Once getting into rhythm with the variety of options I could add into the log, I came up with a few extras in my head, then wrote out this second bit of scribble, which was the foundation of the terminal application.

I wanted to add these four key features → with no errors, make it look clean and easy to read, and all functions to work as they say.



TRAVEL LOG

TRAVEL DIRECTORY

~~Leader board~~
- Add previous Travels
- Edit previous travels
- Delete previous travels.

LEADERBOARD

- Show rankins from Best to worst on a seperate directory.
- Easy to read.

INFO PANEL.

- Different way of displaying info from travels.

WISHLIST

- plan future Travels with, by adding it to the wishlist

show list of wishes.

TL
Directory
Other
Lederboard
Add, edit, delete
Wishlist
View
add
RANkins system
more info panel
updated wishlist

1. Want the application to look clean and easy to read. .
2. Zero errors.
3. All functions work as they should..

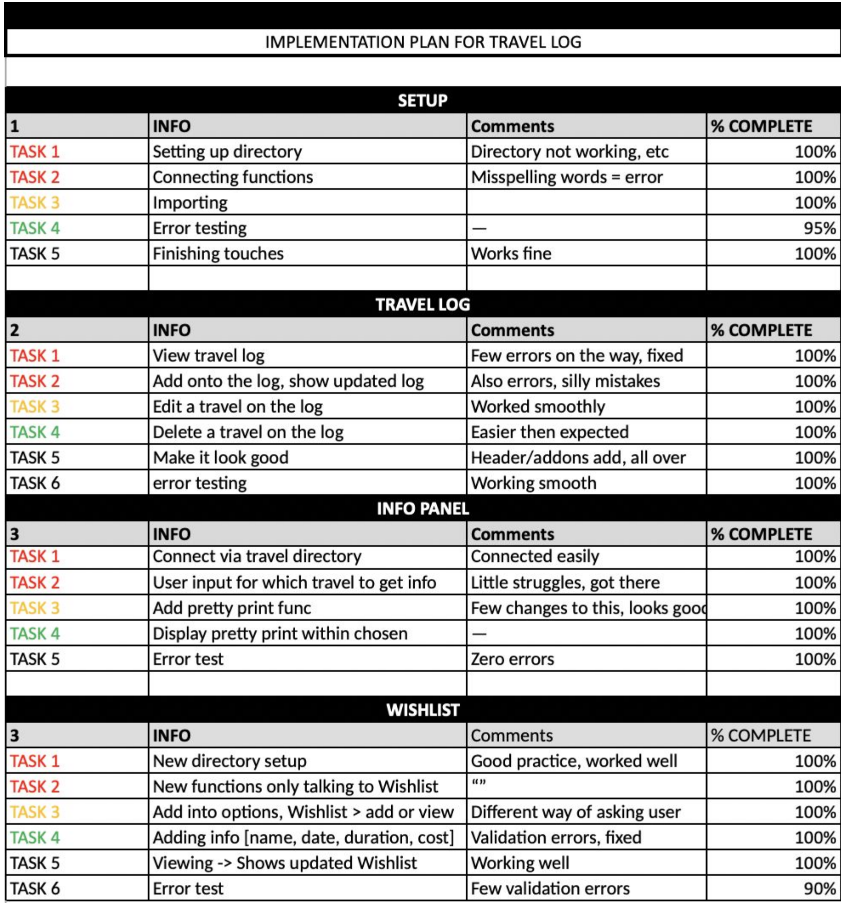
TRAVEL LOG

- An easy reading application that you can add, edit, or delete future or previous travel locations.

- A travel log of all your previous travels. (Where you went, what date it was, how long the trip was, additional comments, and the rating of the trip)

- A leaderboard ranking system, which ranks the highest to lowest scored rating per travel on a separate directory

- Wishlist directory with a viewing board and add function



IMPLEMENTATION PLAN FOR TRAVEL LOG

SETUP

1	INFO	Comments	% COMPLETE
TASK 1	Setting up directory	Directory not working, etc	100%
TASK 2	Connecting functions	Misspelling words = error	100%
TASK 3	Importing		100%
TASK 4	Error testing	—	95%
TASK 5	Finishing touches	Works fine	100%

TRAVEL LOG

2	INFO	Comments	% COMPLETE
TASK 1	View travel log	Few errors on the way, fixed	100%
TASK 2	Add onto the log, show updated log	Also errors, silly mistakes	100%
TASK 3	Edit a travel on the log	Worked smoothly	100%
TASK 4	Delete a travel on the log	Easier then expected	100%
TASK 5	Make it look good	Header/addons add, all over	100%
TASK 6	error testing	Working smooth	100%

INFO PANEL

3	INFO	Comments	% COMPLETE
TASK 1	Connect via travel directory	Connected easily	100%
TASK 2	User input for which travel to get info	Little struggles, got there	100%
TASK 3	Add pretty print func	Few changes to this, looks good	100%
TASK 4	Display pretty print within chosen	—	100%
TASK 5	Error test	Zero errors	100%

WISHLIST

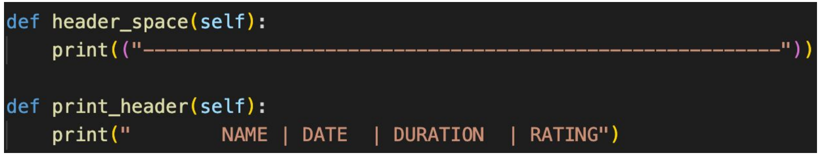
3	INFO	Comments	% COMPLETE
TASK 1	New directory setup	Good practice, worked well	100%
TASK 2	New functions only talking to Wishlist	“”	100%
TASK 3	Add into options, Wishlist > add or view	Different way of asking user	100%
TASK 4	Adding info [name, date, duration, cost]	Validation errors, fixed	100%
TASK 5	Viewing -> Shows updated Wishlist	Working well	100%
TASK 6	Error test	Few validation errors	90%

TRAVEL DIRECTORY

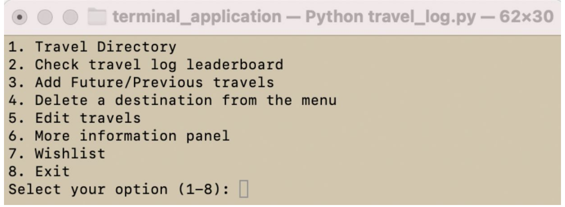
Once inside the directory you can see it's in the format of Name, date, duration, rating.

Once a new travel is added its added to the bottom of the list.

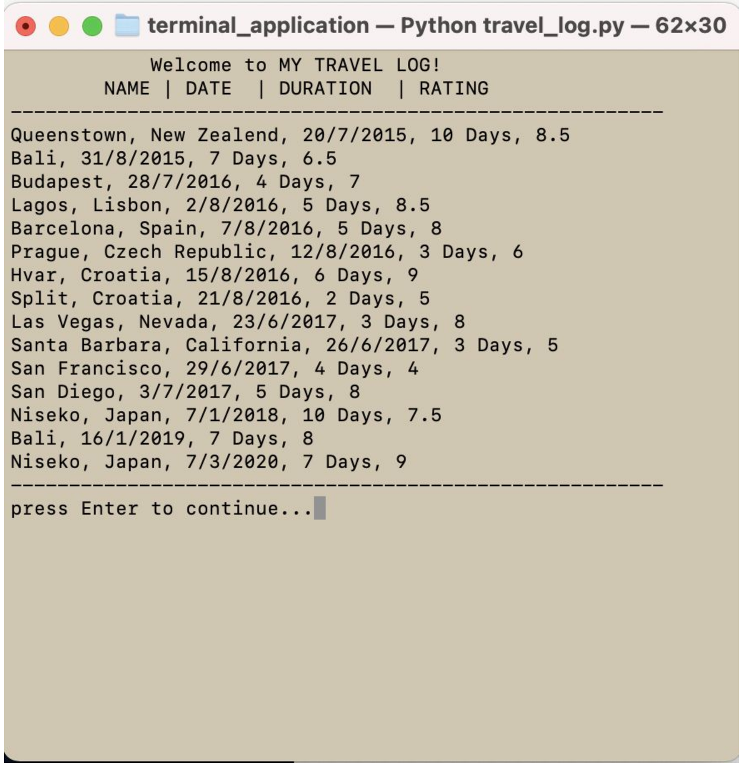
To make all of the directories easy to read, I added two functions (header_space, print_header)



```python
def header_space(self):
    print(("---------------------------------------------------------"))

def print_header(self):
    print("          NAME | DATE  | DURATION  | RATING")
```



```
1. Travel Directory
2. Check travel log leaderboard
3. Add Future/Previous travels
4. Delete a destination from the menu
5. Edit travels
6. More information panel
7. Wishlist
8. Exit
Select your option (1-8): 
```



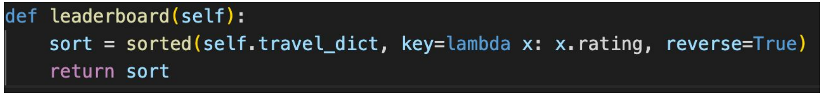
```
            Welcome to MY TRAVEL LOG!
        NAME | DATE  | DURATION  | RATING
---------------------------------------------------------
Queenstown, New Zealand, 20/7/2015, 10 Days, 8.5
Bali, 31/8/2015, 7 Days, 6.5
Budapest, 28/7/2016, 4 Days, 7
Lagos, Lisbon, 2/8/2016, 5 Days, 8.5
Barcelona, Spain, 7/8/2016, 5 Days, 8
Prague, Czech Republic, 12/8/2016, 3 Days, 6
Hvar, Croatia, 15/8/2016, 6 Days, 9
Split, Croatia, 21/8/2016, 2 Days, 5
Las Vegas, Nevada, 23/6/2017, 3 Days, 8
Santa Barbara, California, 26/6/2017, 3 Days, 5
San Francisco, 29/6/2017, 4 Days, 4
San Diego, 3/7/2017, 5 Days, 8
Niseko, Japan, 7/1/2018, 10 Days, 7.5
Bali, 16/1/2019, 7 Days, 8
Niseko, Japan, 7/3/2020, 7 Days, 9
---------------------------------------------------------
press Enter to continue...
```

LEADERBOARD

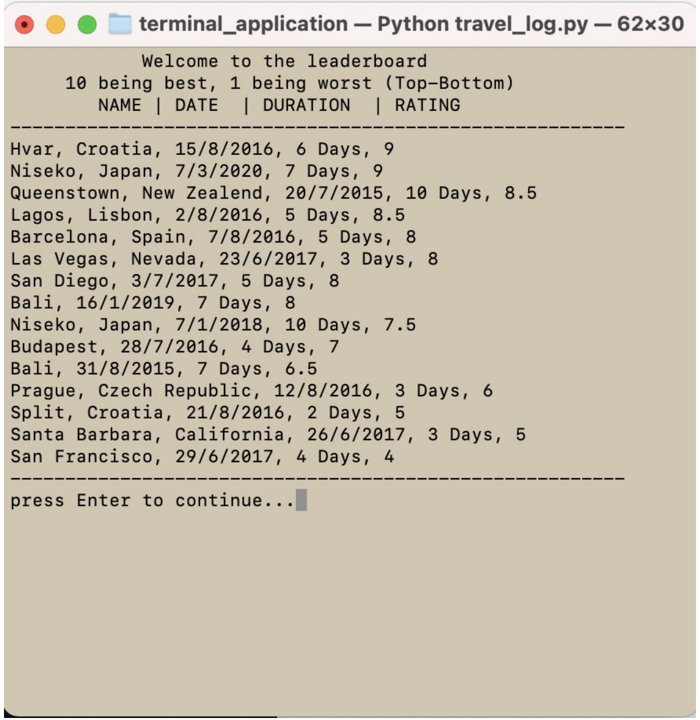
The leaderboard is a rank rating system that shows the best-> worst going off the rating that has been added.

Had a few dramas getting this to work at the start with which part of the argument it would pickup to rate, because initially I had all of the arguments as strings, so it was a little confusing on which part or how to rate.

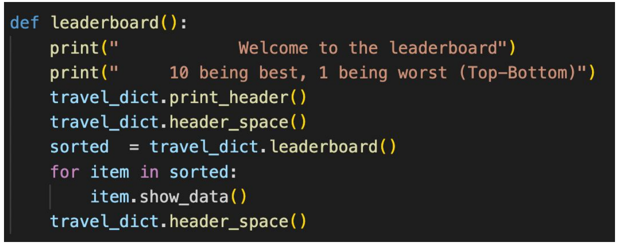
An easy fix was to just change the last argument to a integer, and ask for a float. No longer looks to rate which string, just goes straight to the rating.



```python
def leaderboard(self):
    sort = sorted(self.travel_dict, key=lambda x: x.rating, reverse=True)
    return sort
```



```
            Welcome to the leaderboard
     10 being best, 1 being worst (Top-Bottom)
        NAME | DATE  | DURATION  | RATING
--------------------------------------------------------
Hvar, Croatia, 15/8/2016, 6 Days, 9
Niseko, Japan, 7/3/2020, 7 Days, 9
Queenstown, New Zealend, 20/7/2015, 10 Days, 8.5
Lagos, Lisbon, 2/8/2016, 5 Days, 8.5
Barcelona, Spain, 7/8/2016, 5 Days, 8
Las Vegas, Nevada, 23/6/2017, 3 Days, 8
San Diego, 3/7/2017, 5 Days, 8
Bali, 16/1/2019, 7 Days, 8
Niseko, Japan, 7/1/2018, 10 Days, 7.5
Budapest, 28/7/2016, 4 Days, 7
Bali, 31/8/2015, 7 Days, 6.5
Prague, Czech Republic, 12/8/2016, 3 Days, 6
Split, Croatia, 21/8/2016, 2 Days, 5
Santa Barbara, California, 26/6/2017, 3 Days, 5
San Francisco, 29/6/2017, 4 Days, 4
--------------------------------------------------------
press Enter to continue...
```

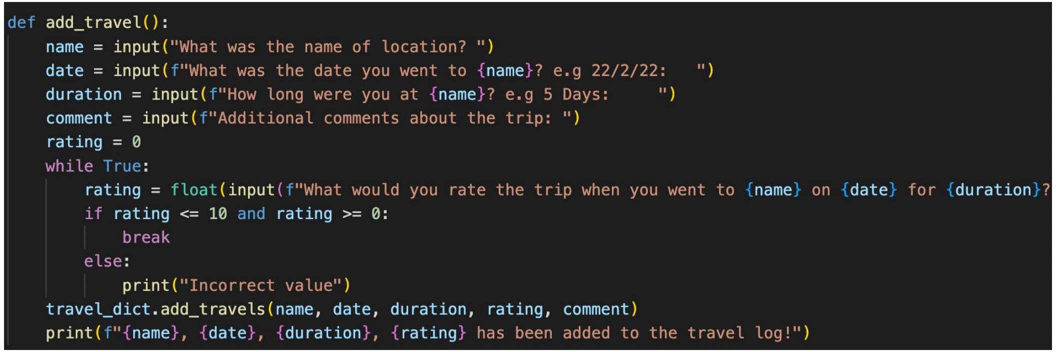


```python
def leaderboard():
    print("            Welcome to the leaderboard")
    print("     10 being best, 1 being worst (Top-Bottom)")
    travel_dict.print_header()
    travel_dict.header_space()
    sorted  = travel_dict.leaderboard()
    for item in sorted:
        item.show_data()
    travel_dict.header_space()
```

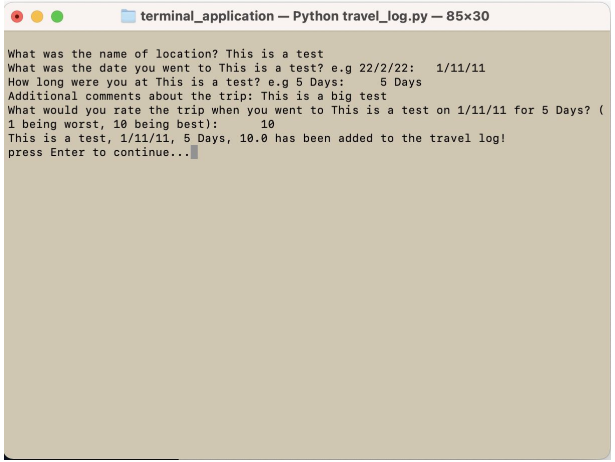
ADDING

I wanted this to look as good as possible, also with a bit of a paper trail of what you've typed in beforehand. As you can see the user input is added into the space where needed.

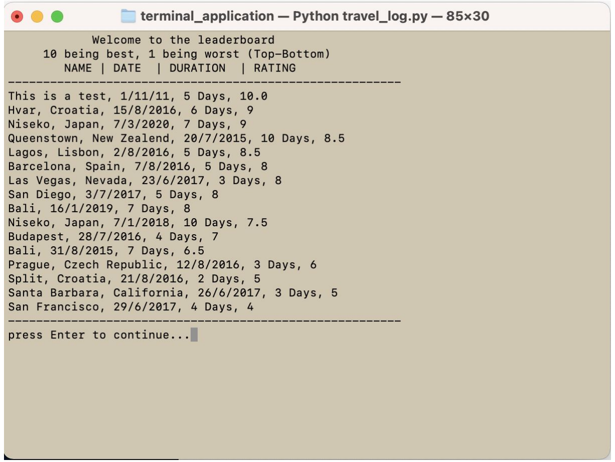
One of the main issues was making sure the user input for the rating was between 1 and 10 to make sure it worked with the leaderboard.



```python
def add_travel():
    name = input("What was the name of location? ")
    date = input(f"What was the date you went to {name}? e.g 22/2/22:   ")
    duration = input(f"How long were you at {name}? e.g 5 Days:     ")
    comment = input(f"Additional comments about the trip: ")
    rating = 0
    while True:
        rating = float(input(f"What would you rate the trip when you went to {name} on {date} for {duration}?
        if rating <= 10 and rating >= 0:
            break
        else:
            print("Incorrect value")
    travel_dict.add_travels(name, date, duration, rating, comment)
    print(f"{name}, {date}, {duration}, {rating} has been added to the travel log!")
```



```
What was the name of location? This is a test
What was the date you went to This is a test? e.g 22/2/22:   1/11/11
How long were you at This is a test? e.g 5 Days:     5 Days
Additional comments about the trip: This is a big test
What would you rate the trip when you went to This is a test on 1/11/11 for 5 Days? (
1 being worst, 10 being best):      10
This is a test, 1/11/11, 5 Days, 10.0 has been added to the travel log!
press Enter to continue...
```

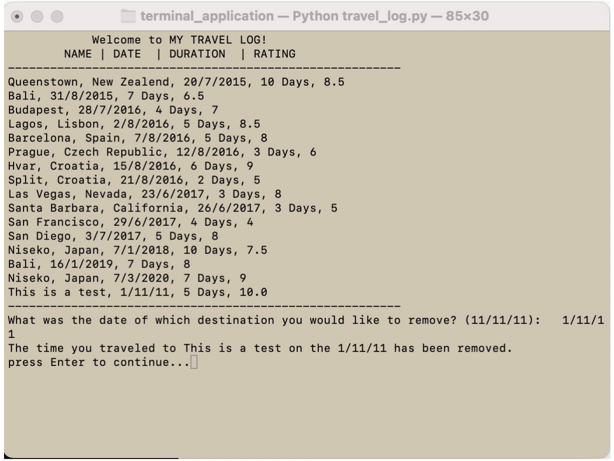


```
            Welcome to the leaderboard
     10 being best, 1 being worst (Top-Bottom)
        NAME | DATE  | DURATION  | RATING
--------------------------------------------------------
This is a test, 1/11/11, 5 Days, 10.0
Hvar, Croatia, 15/8/2016, 6 Days, 9
Niseko, Japan, 7/3/2020, 7 Days, 9
Queenstown, New Zealend, 20/7/2015, 10 Days, 8.5
Lagos, Lisbon, 2/8/2016, 5 Days, 8.5
Barcelona, Spain, 7/8/2016, 5 Days, 8
Las Vegas, Nevada, 23/6/2017, 3 Days, 8
San Diego, 3/7/2017, 5 Days, 8
Bali, 16/1/2019, 7 Days, 8
Niseko, Japan, 7/1/2018, 10 Days, 7.5
Budapest, 28/7/2016, 4 Days, 7
Bali, 31/8/2015, 7 Days, 6.5
Prague, Czech Republic, 12/8/2016, 3 Days, 6
Split, Croatia, 21/8/2016, 2 Days, 5
Santa Barbara, California, 26/6/2017, 3 Days, 5
San Francisco, 29/6/2017, 4 Days, 4
--------------------------------------------------------
press Enter to continue...
```

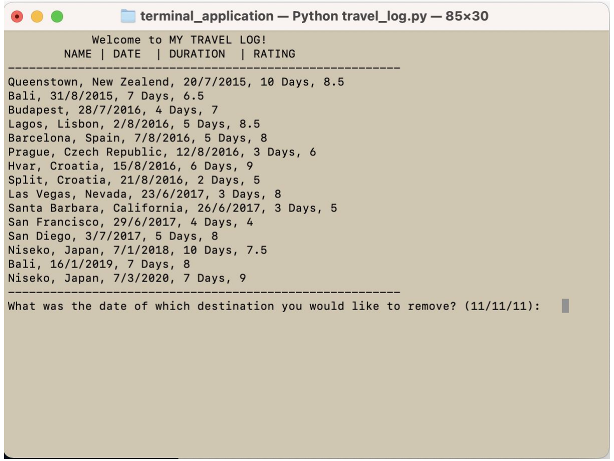
DELETING

Deleting is just done in reverse to adding

The way it works is deleting the item attached to the date. So user input is asking for a date of the trip. To make finding the date easier, the directory is printed out above.



```
            Welcome to MY TRAVEL LOG!
        NAME | DATE  | DURATION  | RATING
--------------------------------------------------------
Queenstown, New Zealend, 20/7/2015, 10 Days, 8.5
Bali, 31/8/2015, 7 Days, 6.5
Budapest, 28/7/2016, 4 Days, 7
Lagos, Lisbon, 2/8/2016, 5 Days, 8.5
Barcelona, Spain, 7/8/2016, 5 Days, 8
Prague, Czech Republic, 12/8/2016, 3 Days, 6
Hvar, Croatia, 15/8/2016, 6 Days, 9
Split, Croatia, 21/8/2016, 2 Days, 5
Las Vegas, Nevada, 23/6/2017, 3 Days, 8
Santa Barbara, California, 26/6/2017, 3 Days, 5
San Francisco, 29/6/2017, 4 Days, 4
San Diego, 3/7/2017, 5 Days, 8
Niseko, Japan, 7/1/2018, 10 Days, 7.5
Bali, 16/1/2019, 7 Days, 8
Niseko, Japan, 7/3/2020, 7 Days, 9
This is a test, 1/11/11, 5 Days, 10.0
--------------------------------------------------------
What was the date of which destination you would like to remove? (11/11/11):   1/11/1
1
The time you traveled to This is a test on the 1/11/11 has been removed.
press Enter to continue...
```



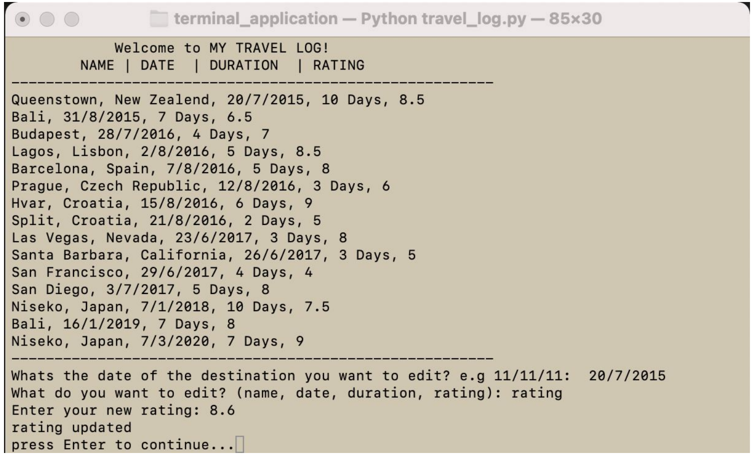
```
            Welcome to MY TRAVEL LOG!
        NAME | DATE  | DURATION  | RATING
--------------------------------------------------------
Queenstown, New Zealend, 20/7/2015, 10 Days, 8.5
Bali, 31/8/2015, 7 Days, 6.5
Budapest, 28/7/2016, 4 Days, 7
Lagos, Lisbon, 2/8/2016, 5 Days, 8.5
Barcelona, Spain, 7/8/2016, 5 Days, 8
Prague, Czech Republic, 12/8/2016, 3 Days, 6
Hvar, Croatia, 15/8/2016, 6 Days, 9
Split, Croatia, 21/8/2016, 2 Days, 5
Las Vegas, Nevada, 23/6/2017, 3 Days, 8
Santa Barbara, California, 26/6/2017, 3 Days, 5
San Francisco, 29/6/2017, 4 Days, 4
San Diego, 3/7/2017, 5 Days, 8
Niseko, Japan, 7/1/2018, 10 Days, 7.5
Bali, 16/1/2019, 7 Days, 8
Niseko, Japan, 7/3/2020, 7 Days, 9
--------------------------------------------------------
What was the date of which destination you would like to remove? (11/11/11):
```

EDITING

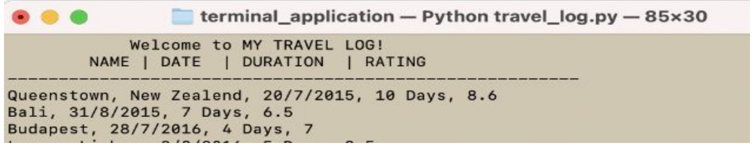
Once entering the edit function, you will be prompted with the travel directory to see which travel you'd like to edit. Same function of picking the date for the same reasons as above.

Once picking the date of which travel you'd like to edit, another prompt will ask for user input on what part of the item you'd like to edit. (Name, date, duration, rating).

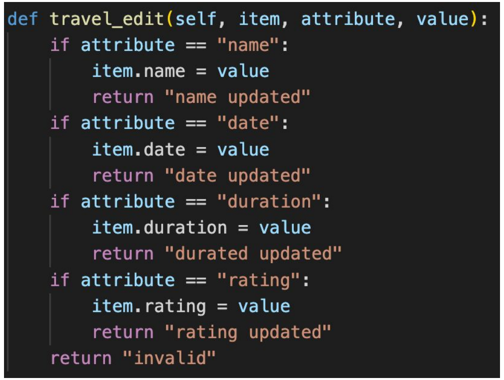
To follow the example, I'm going to edit the rating. After typing in 'rating' another user input will prompt asking what you would like to change the rating too. After this the item will be updated into the directory



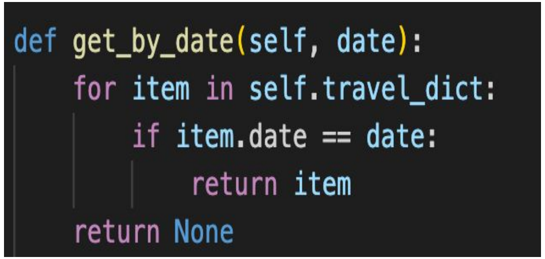
```
               Welcome to MY TRAVEL LOG!
           NAME | DATE   | DURATION   | RATING
--------------------------------------------------------
Queenstown, New Zealend, 20/7/2015, 10 Days, 8.5
Bali, 31/8/2015, 7 Days, 6.5
Budapest, 28/7/2016, 4 Days, 7
Lagos, Lisbon, 2/8/2016, 5 Days, 8.5
Barcelona, Spain, 7/8/2016, 5 Days, 8
Prague, Czech Republic, 12/8/2016, 3 Days, 6
Hvar, Croatia, 15/8/2016, 6 Days, 9
Split, Croatia, 21/8/2016, 2 Days, 5
Las Vegas, Nevada, 23/6/2017, 3 Days, 8
Santa Barbara, California, 26/6/2017, 3 Days, 5
San Francisco, 29/6/2017, 4 Days, 4
San Diego, 3/7/2017, 5 Days, 8
Niseko, Japan, 7/1/2018, 10 Days, 7.5
Bali, 16/1/2019, 7 Days, 8
Niseko, Japan, 7/3/2020, 7 Days, 9
--------------------------------------------------------
Whats the date of the destination you want to edit? e.g 11/11/11:  20/7/2015
What do you want to edit? (name, date, duration, rating): rating
Enter your new rating: 8.6
rating updated
press Enter to continue...
```



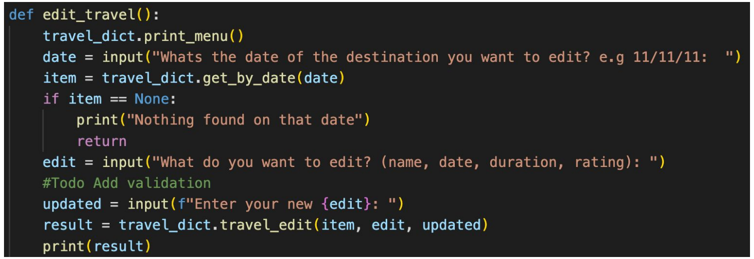
```
               Welcome to MY TRAVEL LOG!
           NAME | DATE   | DURATION   | RATING
--------------------------------------------------------
Queenstown, New Zealend, 20/7/2015, 10 Days, 8.6
Bali, 31/8/2015, 7 Days, 6.5
Budapest, 28/7/2016, 4 Days, 7
```



```python
def travel_edit(self, item, attribute, value):
    if attribute == "name":
        item.name = value
        return "name updated"
    if attribute == "date":
        item.date = value
        return "date updated"
    if attribute == "duration":
        item.duration = value
        return "durated updated"
    if attribute == "rating":
        item.rating = value
        return "rating updated"
    return "invalid"
```



```python
def get_by_date(self, date):
    for item in self.travel_dict:
        if item.date == date:
            return item
    return None
```



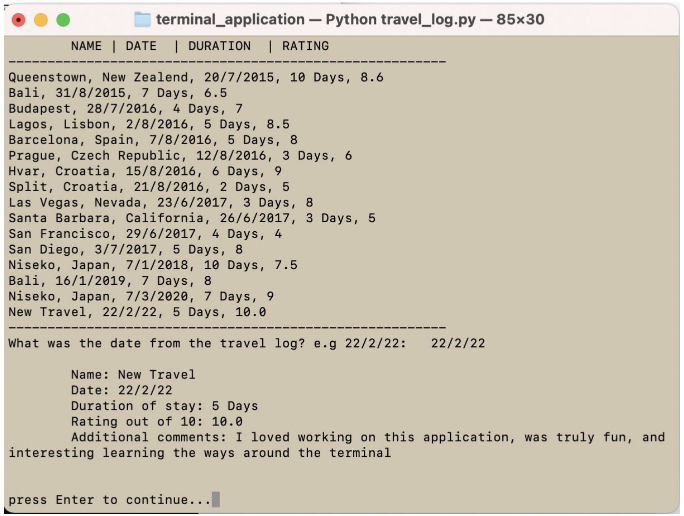
```python
def edit_travel():
    travel_dict.print_menu()
    date = input("Whats the date of the destination you want to edit? e.g 11/11/11:  ")
    item = travel_dict.get_by_date(date)
    if item == None:
        print("Nothing found on that date")
        return
    edit = input("What do you want to edit? (name, date, duration, rating): ")
    #Todo Add validation
    updated = input(f"Enter your new {edit}: ")
    result = travel_dict.travel_edit(item, edit, updated)
    print(result)
```

INFO PANEL

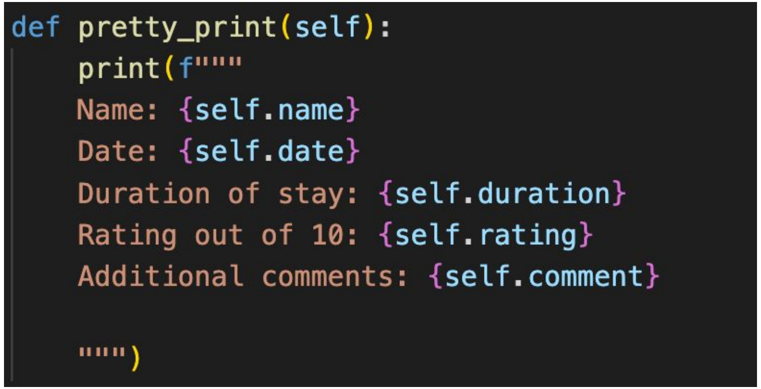
Once through the menu prompts the "More information panel" where you can view the travel in a more direct aspect and view the comments added.

The application will prompt you with the travel directory, for easy choosing of which to pick. As a theme you will be choosing the destination via the date.

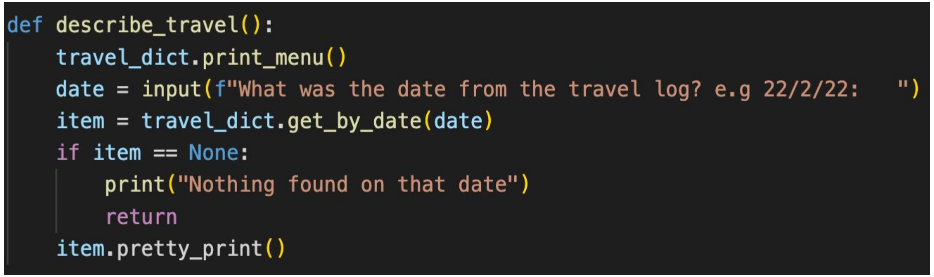
User input asked for "What was the date from the travel log?"



```
        NAME | DATE   | DURATION   | RATING
--------------------------------------------------------
Queenstown, New Zealend, 20/7/2015, 10 Days, 8.6
Bali, 31/8/2015, 7 Days, 6.5
Budapest, 28/7/2016, 4 Days, 7
Lagos, Lisbon, 2/8/2016, 5 Days, 8.5
Barcelona, Spain, 7/8/2016, 5 Days, 8
Prague, Czech Republic, 12/8/2016, 3 Days, 6
Hvar, Croatia, 15/8/2016, 6 Days, 9
Split, Croatia, 21/8/2016, 2 Days, 5
Las Vegas, Nevada, 23/6/2017, 3 Days, 8
Santa Barbara, California, 26/6/2017, 3 Days, 5
San Francisco, 29/6/2017, 4 Days, 4
San Diego, 3/7/2017, 5 Days, 8
Niseko, Japan, 7/1/2018, 10 Days, 7.5
Bali, 16/1/2019, 7 Days, 8
Niseko, Japan, 7/3/2020, 7 Days, 9
New Travel, 22/2/22, 5 Days, 10.0
--------------------------------------------------------
What was the date from the travel log? e.g 22/2/22:   22/2/22

        Name: New Travel
        Date: 22/2/22
        Duration of stay: 5 Days
        Rating out of 10: 10.0
        Additional comments: I loved working on this application, was truly fun, and
interesting learning the ways around the terminal


press Enter to continue...
```



```python
def pretty_print(self):
    print(f"""
    Name: {self.name}
    Date: {self.date}
    Duration of stay: {self.duration}
    Rating out of 10: {self.rating}
    Additional comments: {self.comment}

    """)
```



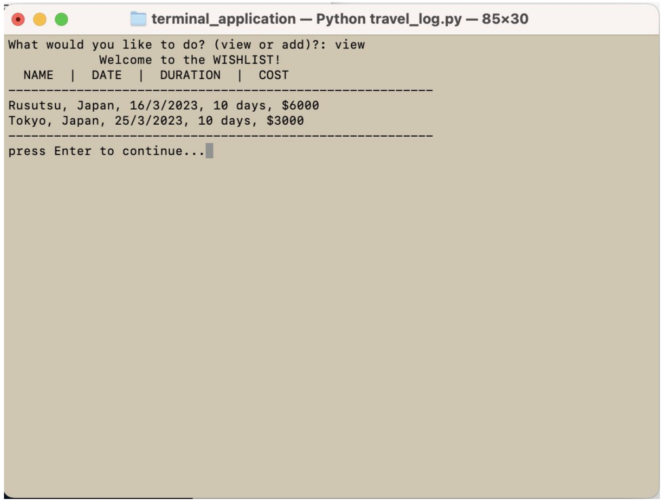
```python
def describe_travel():
    travel_dict.print_menu()
    date = input(f"What was the date from the travel log? e.g 22/2/22:    ")
    item = travel_dict.get_by_date(date)
    if item == None:
        print("Nothing found on that date")
        return
    item.pretty_print()
```

WISHLIST

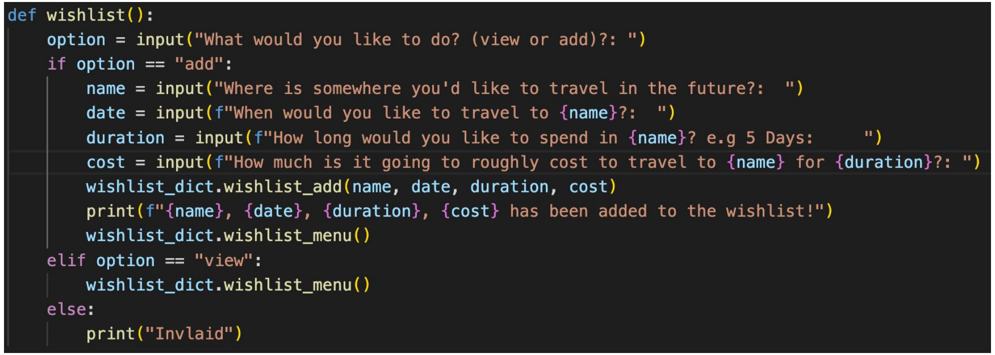
The wishlist function is there to add future trips, to a seperate directory. This has got nothing connected with the Travel directory.

Once you prompt the wishlist with the "7" key you are asked by the application for input on which you'd rather do;

- **view:** Shows all the items on your wishlist. The name, the date you'd like to travel, days potential, and how much it’s going to cost.



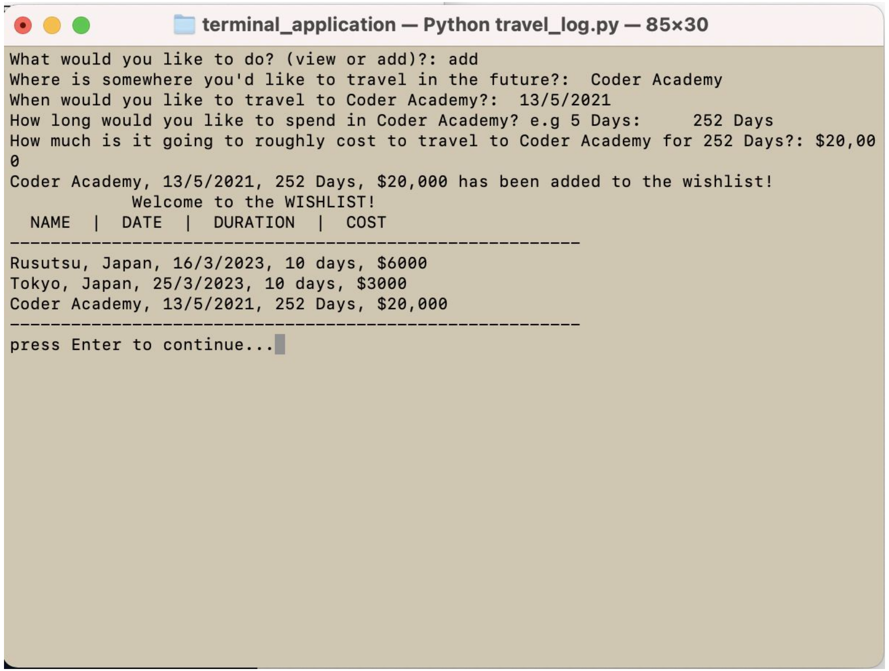
```
What would you like to do? (view or add)?: view
            Welcome to the WISHLIST!
  NAME  |  DATE  |  DURATION  |  COST
--------------------------------------------------------
Rusutsu, Japan, 16/3/2023, 10 days, $6000
Tokyo, Japan, 25/3/2023, 10 days, $3000
--------------------------------------------------------
press Enter to continue...
```



```python
def wishlist():
    option = input("What would you like to do? (view or add)?: ")
    if option == "add":
        name = input("Where is somewhere you'd like to travel in the future?:  ")
        date = input(f"When would you like to travel to {name}?:  ")
        duration = input(f"How long would you like to spend in {name}? e.g 5 Days:     ")
        cost = input(f"How much is it going to roughly cost to travel to {name} for {duration}?: ")
        wishlist_dict.wishlist_add(name, date, duration, cost)
        print(f"{name}, {date}, {duration}, {cost} has been added to the wishlist!")
        wishlist_dict.wishlist_menu()
    elif option == "view":
        wishlist_dict.wishlist_menu()
    else:
        print("Invlaid")
```

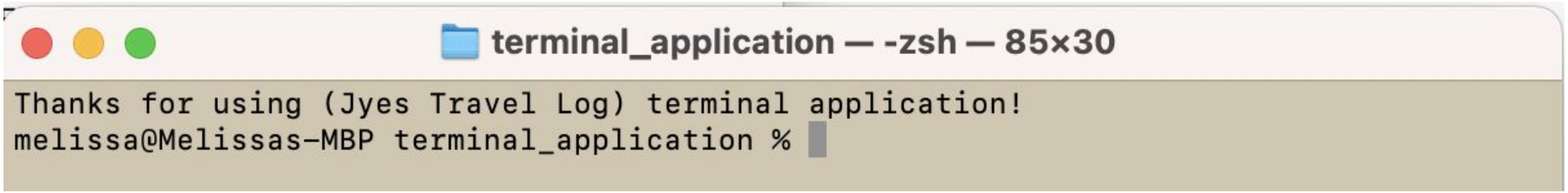
WISHLIST

- **add:** prompts you for input. Starting off with the name of the destination you'd like to travel.
- After adding the destination, user input will be prompted again with the application asking for the date you'd like to travel to this location.
- After adding the date, another user input will be prompted asking for the duration, and then again with the cost.
- After hitting the 'Enter' key, the application will confirm the details and publish it into the Wishlist.



```
What would you like to do? (view or add)?: add
Where is somewhere you'd like to travel in the future?:  Coder Academy
When would you like to travel to Coder Academy?:  13/5/2021
How long would you like to spend in Coder Academy? e.g 5 Days:     252 Days
How much is it going to roughly cost to travel to Coder Academy for 252 Days?: $20,00
0
Coder Academy, 13/5/2021, 252 Days, $20,000 has been added to the wishlist!
            Welcome to the WISHLIST!
  NAME  |  DATE  |  DURATION  |  COST
--------------------------------------------------------
Rusutsu, Japan, 16/3/2023, 10 days, $6000
Tokyo, Japan, 25/3/2023, 10 days, $3000
Coder Academy, 13/5/2021, 252 Days, $20,000
--------------------------------------------------------
press Enter to continue...
```

END



```
Thanks for using (Jyes Travel Log) terminal application!
melissa@Melissas-MBP terminal_application %
```