CAPSTONE PROJECT:--------------------------

PROJECT 9:- TASK MANAGER

TEAM (ASAN)

CONTRIBUTORS:-

NEELABH RANA(202301476)

SOHAM MEVADA(202301484)

ATIK VOHRA(202301447)

AKSHAT BHATT(202301460)

Github link:- https://github.com/Neelabh1929/Capstone.git

GUI LINK:https: //atikdagu.github.io/todoweb/

1.) Implementation using linked list

- We have used linked list to implement our project because it is very easy and fast to implement. And also we can store our elements in a linear way , so the traversal becomes much more easier , and we can have a fast access to things.

- We have added a priority function which checks the date of the task and puts it accordingly in the list. Which is similar to the priority queue.

PSEUDOCODE FOR IMPLEMENTATION USING LINKED LIST

1 Class Definitions

1.1 Node_daily Class

```
class Node_daily:
    string type, message, deadline
    ll imp_lvl
    Node_daily *next
    Node_daily *prev

    constructor Node_daily(s):
        type = s
        message = ""
        imp_lvl = 0
        deadline = ""
        next = NULL
        prev = NULL

    constructor Node_daily(s, m, a, deadlinetime):
        type = s
        message = m
        imp_lvl = a
        deadline = deadlinetime
        next = NULL
        prev = NULL

    friend function compare_time_daily(string)
    friend function insert_daily_task()
    friend function display_daily()
    friend function display_incomplete_dailytask()
    friend function delete_daily_task()
    friend function write_daily_task(string)
    friend function load_daily_task_file(string)
    friend function erase_all_daily_task(string)
    friend function insert_daily(deadlinetime, type
    message, imp_lvl)

Node_daily *head_daily = new Node_daily("Daily Task")
```

1.2 Node Class

```
class Node:
    string time, time1, type, message
    ll imp_lvl
    time = t
    time1 = t1
    type = type1
    message = message1
    imp_lvl = i
    next = NULL
    prev = NULL

constructor Node(t, t1, type1, message1):
    time = t
    time1 = t1
    type = type1
    message = message1
    imp_lvl = 0
    next = NULL
    prev = NULL

constructor Node(type1, message1):
    type = type1
    message = message1
    imp_lvl = 0
    next = NULL
    prev = NULL

constructor Node(type1, i):
    time = ""
    type = type1
    message = ""
    imp_lvl = i
    next = NULL
    prev = NULL

constructor Node(type1):
    type = type1
    next = NULL
    prev = NULL

friend function insert_task_common(string, string,
   ↪ string, string, ll)
friend function createlist()
friend function priority(Node *, Node *)
friend function print_common_list(Node *head_common)
friend function writeDataToFile(string)
friend function eraseFileContents(const string &)
friend function cmoparet2ime(Node *current, ll chour,
   ↪ ll cminute, ll csecond, ll cdate, ll cmonth, ll
   ↪ cyear)
friend function reminder_x_hr(ll)
friend function remove_tasks(string)
friend function alaram(Node *, string)
```

```
    friend function is_list_empty()
    friend function reschedule(string)
    friend function edit_task(string)
    friend function daily_task()
    friend function insert_task_daily()

Node *head_common = new Node("Common task list")
```

2 Helper Functions

2.1 stringToInt

```
 \begin{lstlisting}
stringToInt(str):
    value = 0
    for i = 0 to str.length() - 1:
        c = str[i]
        if c is between '0' and '9':
            value = value * 10 + (c - '0')
        else:
            print "Unexpected character encountered:", c
    return value
```

2.2 extractDateComponents

```
extractDateComponents(date, day, month, year):
    day = stringToInt(date.substr(0, 2))
    month = stringToInt(date.substr(2, 2))
    year = stringToInt(date.substr(4, 4))
```

2.3 extractTimeComponents

```
extractTimeComponents(time, hours, minutes, seconds):
    hours = stringToInt(time.substr(0, 2))
    minutes = stringToInt(time.substr(2, 2))
    seconds = stringToInt(time.substr(4, 2))
```

2.4 correct date

```
correct_date(deadlinedate):
    year, month, date = extractDateComponents(deadlinedate)
```

2.5 correct time

```
correct_time(deadlinetime): hour, minute, second =
extractTimeComponents(deadlinetime)
```

2.6 priority

```
priority(temp, temp_insert):
    Compare task priorities based on their deadlines and
    importance levels
```

2.7 insert task common

```
insert_task_common(deadlinedate, deadlinetime,
type_by_user, message, imp_lvl_by_user):
     Insert a task into the common list based on its
     priority
```

2.8 compare time ctime user time

```
compare_time_ctime_user_time(user_date, user_time):
     Compare user-provided date and time with current date
     and time
```

2.9 insert task

```
insert_task():
     Function to insert a new task based on user input
```

2.10 print common list

```
print_common_list(head):
     Print all tasks in the common list
```

2.11 writeDataToFile

```
writeDataToFile(username):
     Write task data to a file for a specific user
```

2.12 loadDataFromFile

```
loadDataFromFile(username):
    Load task data from a file for a specific user
```

2.13 eraseFileContents

```
eraseFileContents(filename):
     Erase the contents of a file
```

2.14 cmoparet2ime

```
compare2time(current, chour, cminute, csecond, cdate,
   ↪ cmonth, cyear):
    Compare two time instances
```

2.15 checkdate

```
checkdate(month, date, year):
    Check if the given date is valid
```

2.16 is list empty

```
is_list_empty():
     Check if the common list is empty
```

2.17 reminder x hr

```
reminder_x_hr(rhour):
     Print reminders for tasks due within a specified time
```

2.18 remove tasks

```
remove_tasks(username):
     Remove a task from the common list
```

2.19 alaram

```
alaram(head, username):
     Check and print alarms for tasks due
```

2.20 reschedule

```
reschedule(username):
     Reschedule a task
```

2.21 edit task

```
edit_task(username):
    Edit a task
```

2.22 compare time daily

```
compare_time_daily(deadline):
     Compare the deadline of a daily task with current time
```

2.23 insert daily

```
insert_daily(deadlinetime, task_type_user, message,
imp_lvl_user):
     Insert a daily task
```

2.24 insert daily task

```
insert_daily_task():
     Function to insert a new daily task
```

2.25 display daily

```
display_daily():
     Display all daily tasks
```

2.26 display incomplete dailytask

```
display_incomplete_dailytask():
    Display incomplete daily tasks
```

2.27 delete daily task

```
delete_daily_task():
     Delete a daily task
```

2.28 write daily task

```
write_daily_task(username):
     Write daily task data to a file
```

2.29 load daily task file

```
load_daily_task_file(username):
     Load daily task data from a file
```

2.30 erase all daily task

```
erase_all_daily_task(username):
    Erase all daily tasks for a user
```

2.31 loaduname

```
loaduname(unames):
     Load usernames from a file
```

2.32 writeuname

```
writeuname(unames):
    Write usernames to a file
```

2.33 writeuser pass

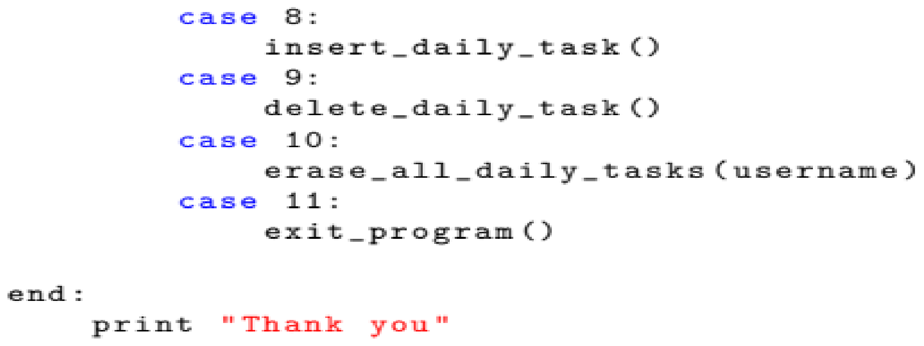
```
writeuser_pass(user_pass):
     Write user passwords to a file
```

2.34 readuser pass

```
readuser_pass(user_pass):
     Read user passwords from a file
```

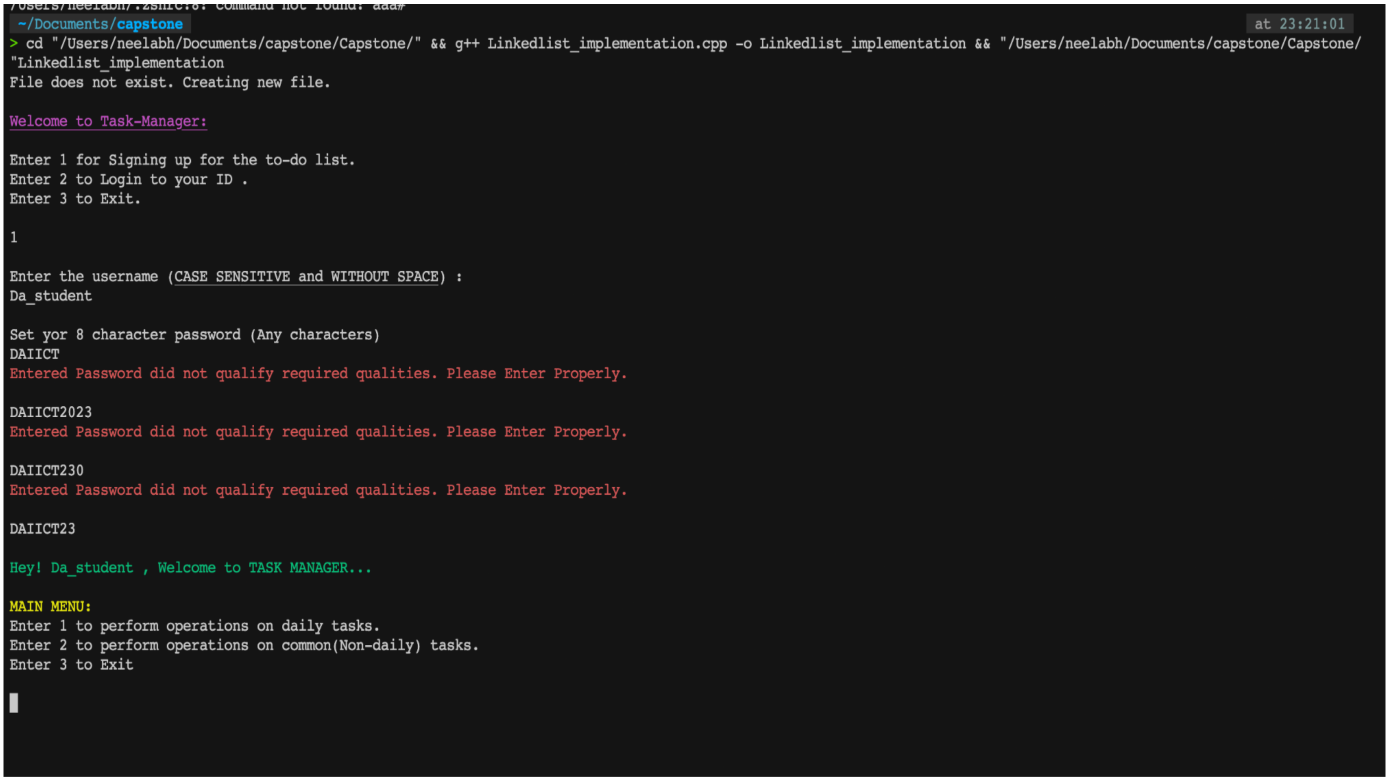
3 Main Function

```
main():
    unames = empty set
    load_unames(unames)
    cnt = size of unames
    user_pass = empty vector of pairs
    read_user_pass(user_pass)

    print "Enter 1 to signup"
    print "Enter 2 to login to your id"
    print "Enter 3 to Exit"
    read user_login

    if user_login == 1:
        signup(unames, user_pass)

    else if user_login == 2:
        login(unames, user_pass)

    else:
        goto end

    write_unames(unames)

    if user_login == 1:
        welcome_message(username)

    load_and_display_tasks(username)

    while true:
        print_menu_options()
        read choice

        switch choice:
            case 1:
                insert_task()
            case 2:
                view_task_list()
            case 3:
                remove_all_tasks(username)
            case 4:
                remove_task(username)
            case 5:
                reschedule_task(username)
            case 6:
                edit_task_details(username)
            case 7:
                display_all_daily_tasks()
```



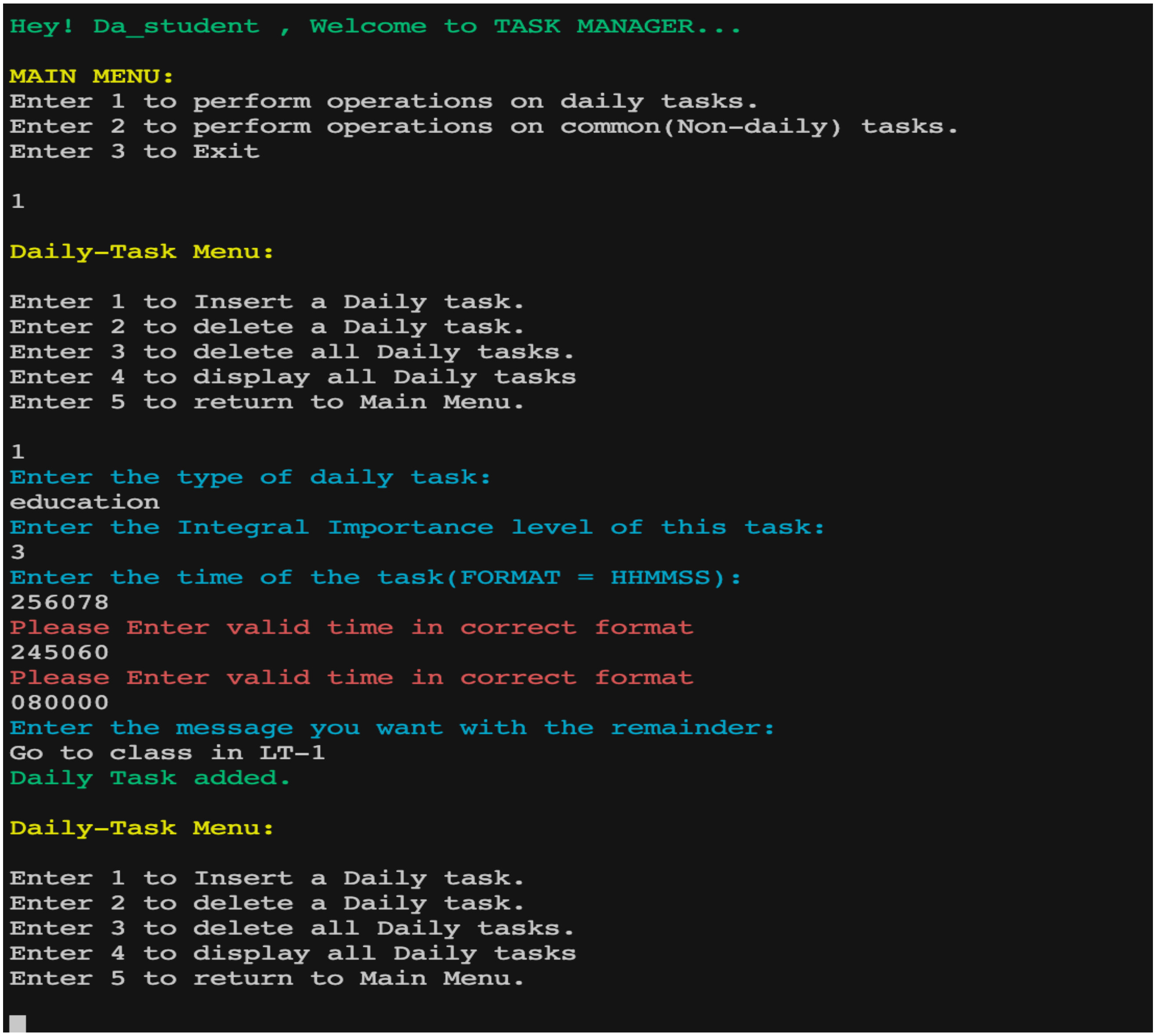
```
            case 8:
                insert_daily_task()
            case 9:
                delete_daily_task()
            case 10:
                erase_all_daily_tasks(username)
            case 11:
                exit_program()

    end:
        print "Thank you"
```

OUTPUT:-



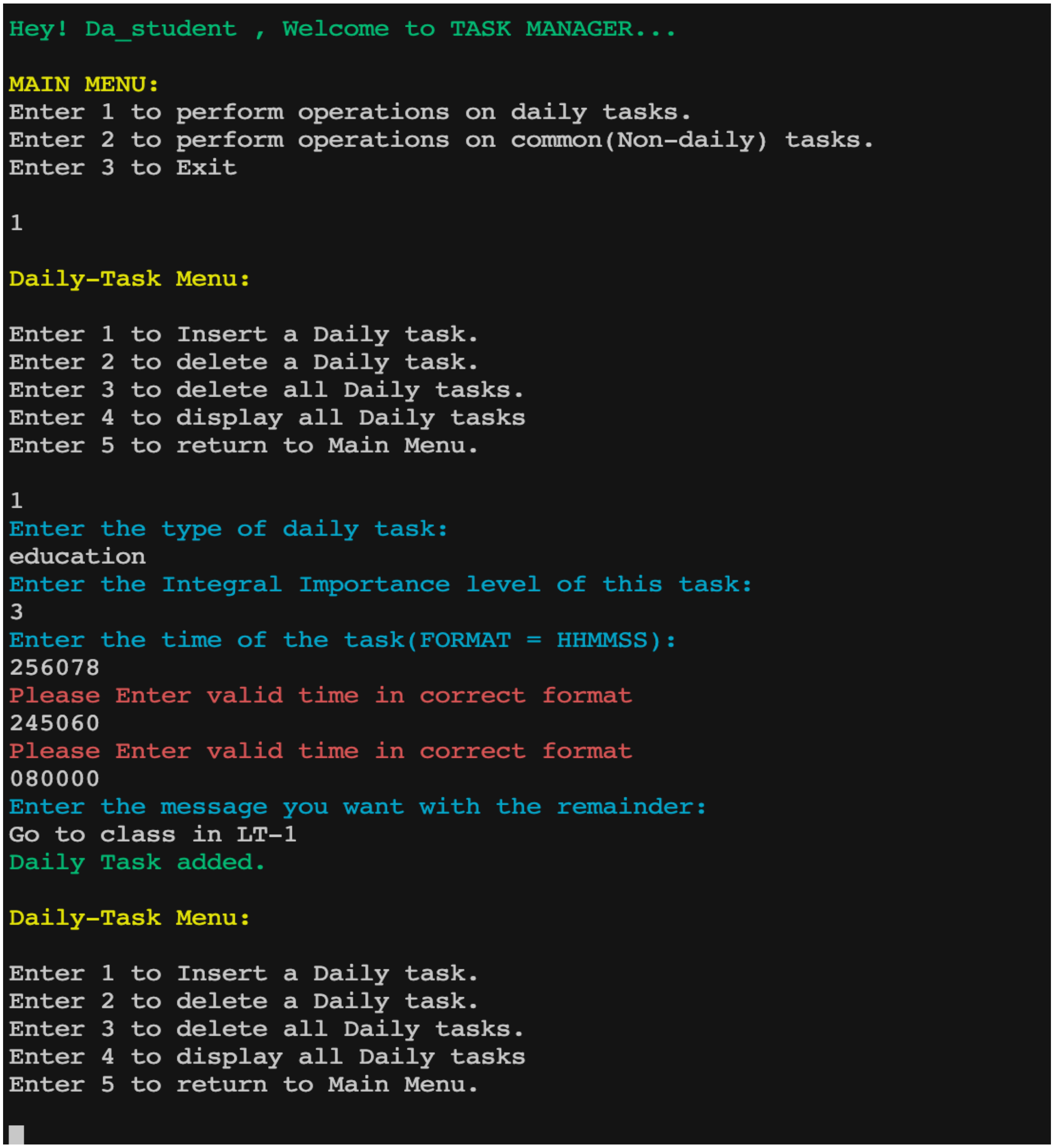
```
/Users/neelabh/.zshrc:8: command not found: aaa#
 ~/Documents/capstone                                                                                                    at 23:21:01
> cd "/Users/neelabh/Documents/capstone/Capstone/" && g++ Linkedlist_implementation.cpp -o Linkedlist_implementation && "/Users/neelabh/Documents/capstone/Capstone/
"Linkedlist_implementation
File does not exist. Creating new file.

Welcome to Task-Manager:

Enter 1 for Signing up for the to-do list.
Enter 2 to Login to your ID .
Enter 3 to Exit.

1

Enter the username (CASE SENSITIVE and WITHOUT SPACE) :
Da_student

Set yor 8 character password (Any characters)
DAIICT
Entered Password did not qualify required qualities. Please Enter Properly.

DAIICT2023
Entered Password did not qualify required qualities. Please Enter Properly.

DAIICT230
Entered Password did not qualify required qualities. Please Enter Properly.

DAIICT23

Hey! Da_student , Welcome to TASK MANAGER...

MAIN MENU:
Enter 1 to perform operations on daily tasks.
Enter 2 to perform operations on common(Non-daily) tasks.
Enter 3 to Exit

▌
```

Here we need to start with signing in or logging in with the username and password.
We signed up and entered username as Da_student. In the password , when we entered password of length other than 8 error statement is displayed , and password is aksed again.
Once we set the password in correct format, we enter to the main menu.



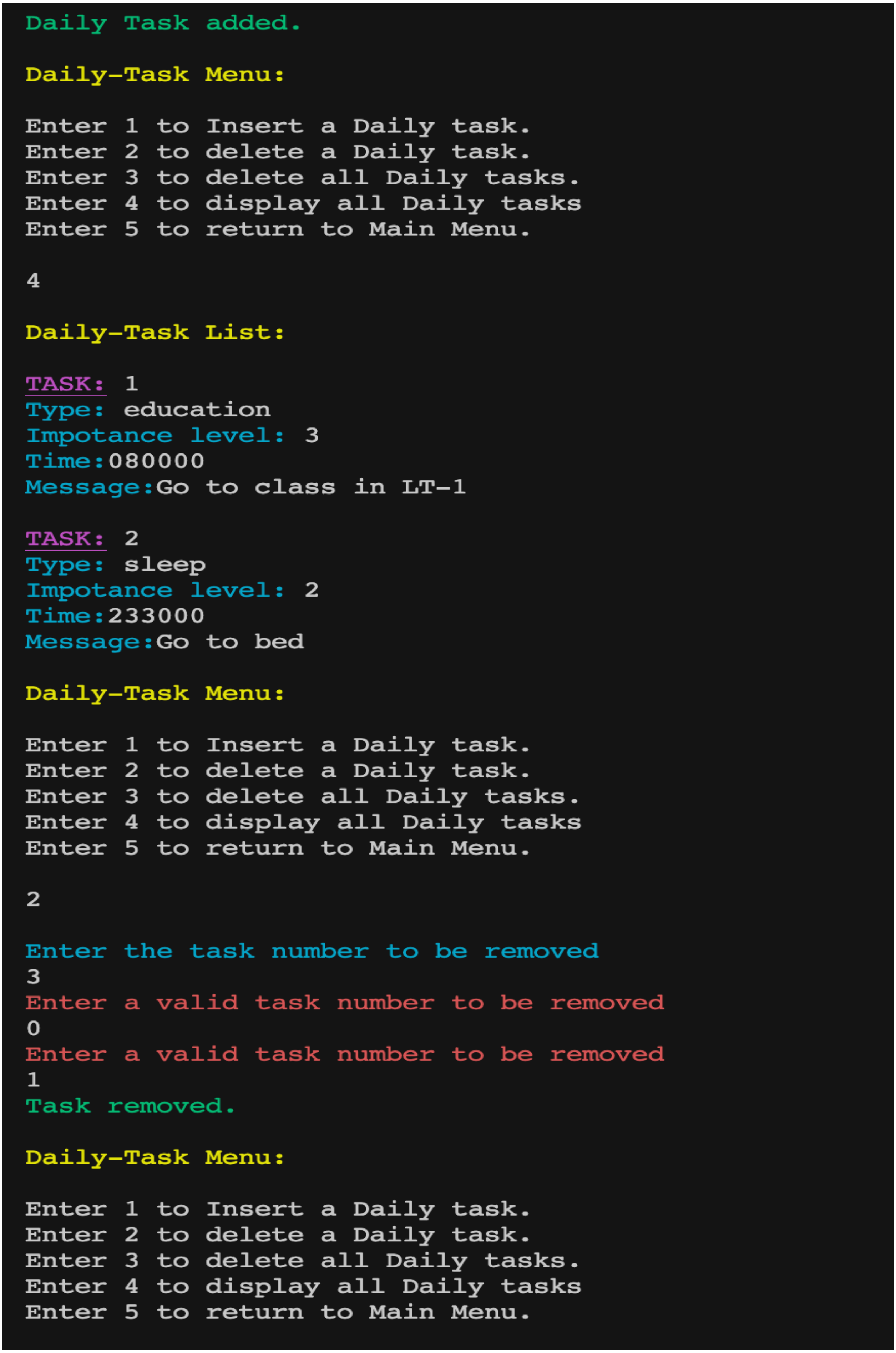
```
Hey! Da_student , Welcome to TASK MANAGER...

MAIN MENU:
Enter 1 to perform operations on daily tasks.
Enter 2 to perform operations on common(Non-daily) tasks.
Enter 3 to Exit

1

Daily-Task Menu:

Enter 1 to Insert a Daily task.
Enter 2 to delete a Daily task.
Enter 3 to delete all Daily tasks.
Enter 4 to display all Daily tasks
Enter 5 to return to Main Menu.

1
Enter the type of daily task:
education
Enter the Integral Importance level of this task:
3
Enter the time of the task(FORMAT = HHMMSS):
256078
Please Enter valid time in correct format
245060
Please Enter valid time in correct format
080000
Enter the message you want with the remainder:
Go to class in LT-1
Daily Task added.

Daily-Task Menu:

Enter 1 to Insert a Daily task.
Enter 2 to delete a Daily task.
Enter 3 to delete all Daily tasks.
Enter 4 to display all Daily tasks
Enter 5 to return to Main Menu.

```

When we enter 1 in main menu , we reach daily menu , and we inserted a daily task. Insertion involves checking of the entered time , if entered time is invalid error is shown.
During insertion, tasks are added in sorted manner as per deadline, deadline time , imp_lvl accordingly.



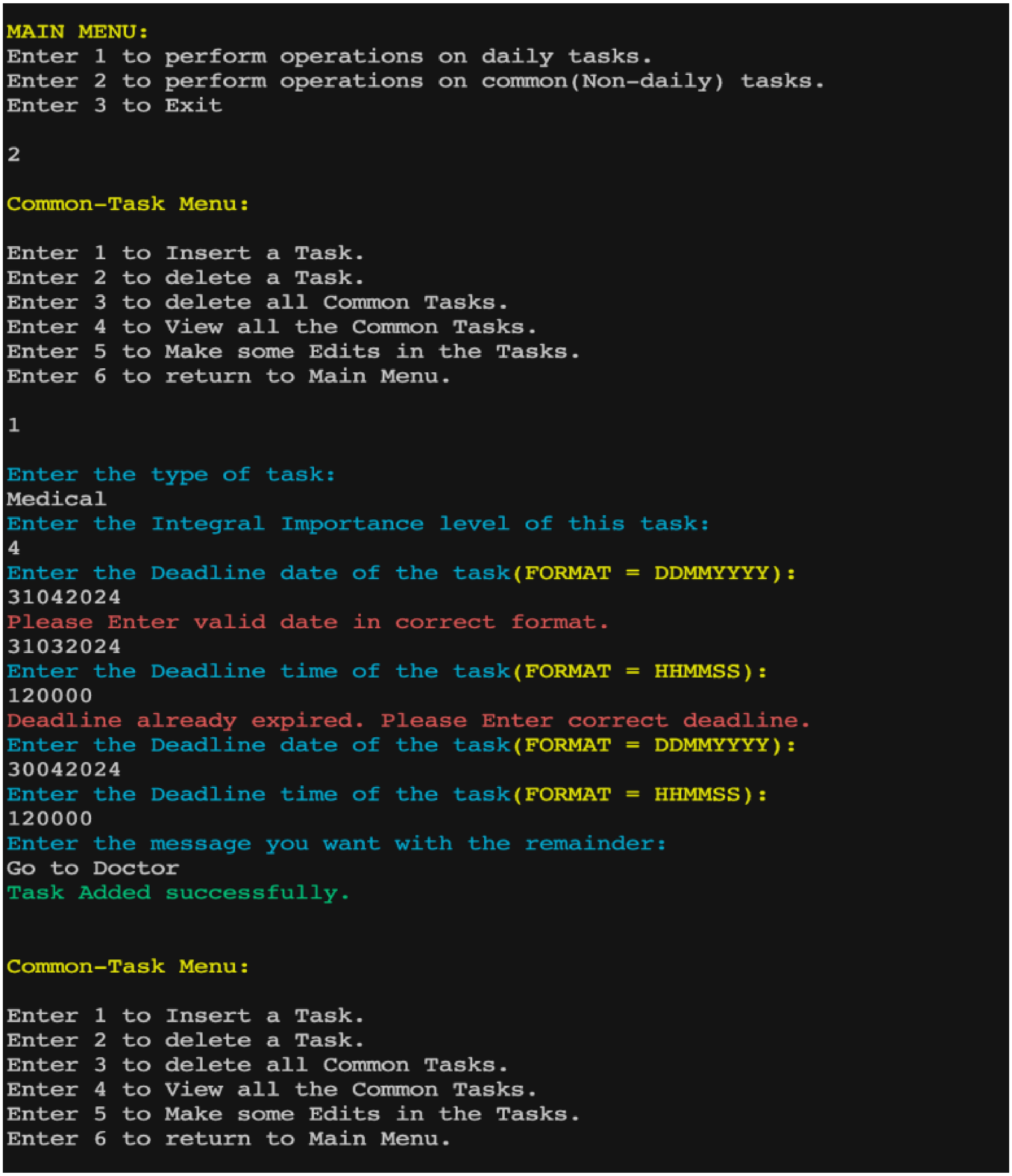
```
Hey! Da_student , Welcome to TASK MANAGER...

MAIN MENU:
Enter 1 to perform operations on daily tasks.
Enter 2 to perform operations on common(Non-daily) tasks.
Enter 3 to Exit

1

Daily-Task Menu:

Enter 1 to Insert a Daily task.
Enter 2 to delete a Daily task.
Enter 3 to delete all Daily tasks.
Enter 4 to display all Daily tasks
Enter 5 to return to Main Menu.

1
Enter the type of daily task:
education
Enter the Integral Importance level of this task:
3
Enter the time of the task(FORMAT = HHMMSS):
256078
Please Enter valid time in correct format
245060
Please Enter valid time in correct format
080000
Enter the message you want with the remainder:
Go to class in LT-1
Daily Task added.

Daily-Task Menu:

Enter 1 to Insert a Daily task.
Enter 2 to delete a Daily task.
Enter 3 to delete all Daily tasks.
Enter 4 to display all Daily tasks
Enter 5 to return to Main Menu.

```

Then we added other daily task.



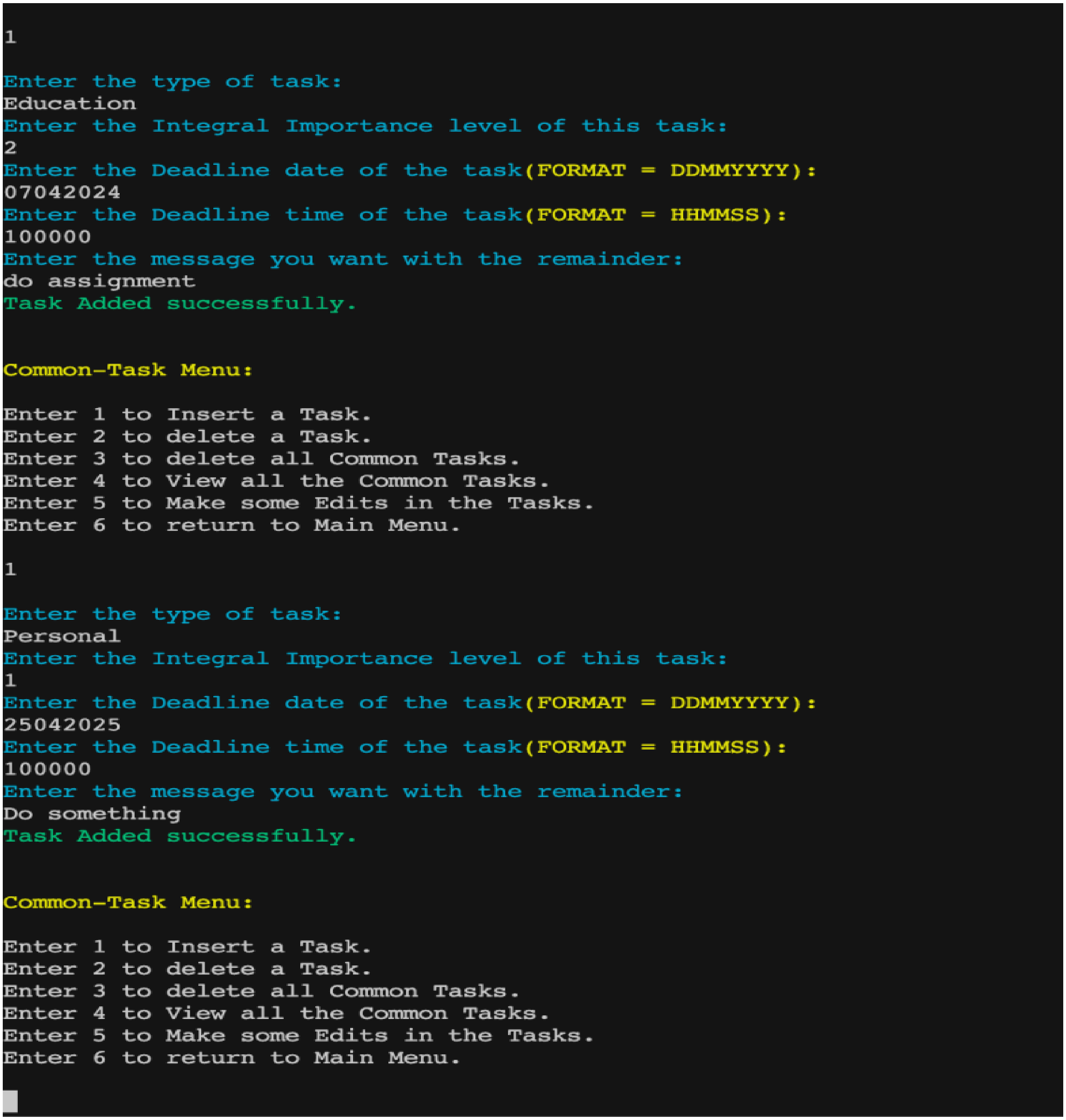
```
Daily Task added.

Daily-Task Menu:

Enter 1 to Insert a Daily task.
Enter 2 to delete a Daily task.
Enter 3 to delete all Daily tasks.
Enter 4 to display all Daily tasks
Enter 5 to return to Main Menu.

4

Daily-Task List:

TASK: 1
Type: education
Impotance level: 3
Time:080000
Message:Go to class in LT-1

TASK: 2
Type: sleep
Impotance level: 2
Time:233000
Message:Go to bed

Daily-Task Menu:

Enter 1 to Insert a Daily task.
Enter 2 to delete a Daily task.
Enter 3 to delete all Daily tasks.
Enter 4 to display all Daily tasks
Enter 5 to return to Main Menu.

2

Enter the task number to be removed
3
Enter a valid task number to be removed
0
Enter a valid task number to be removed
1
Task removed.

Daily-Task Menu:

Enter 1 to Insert a Daily task.
Enter 2 to delete a Daily task.
Enter 3 to delete all Daily tasks.
Enter 4 to display all Daily tasks
Enter 5 to return to Main Menu.
```

Here we can see the whole list of daily task. And we also deleted daily task 1. Code handles for invalid task number also. Same way all daily tasks can also be removed



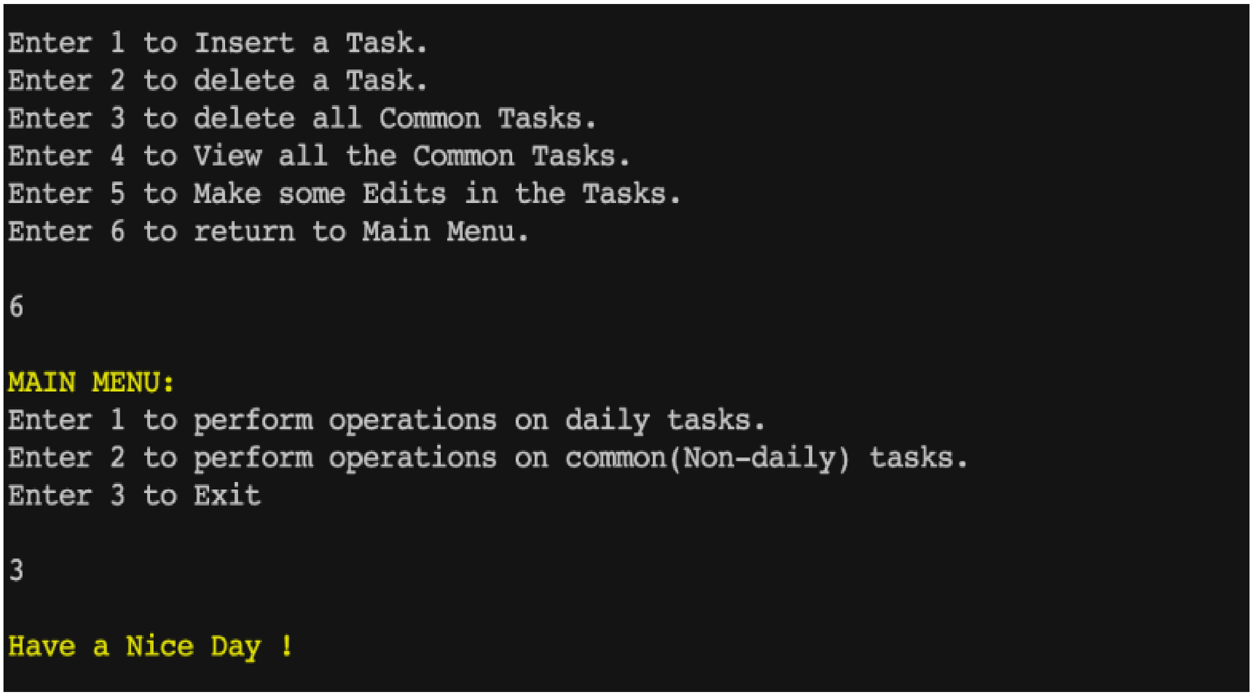
```
MAIN MENU:
Enter 1 to perform operations on daily tasks.
Enter 2 to perform operations on common(Non-daily) tasks.
Enter 3 to Exit

2

Common-Task Menu:

Enter 1 to Insert a Task.
Enter 2 to delete a Task.
Enter 3 to delete all Common Tasks.
Enter 4 to View all the Common Tasks.
Enter 5 to Make some Edits in the Tasks.
Enter 6 to return to Main Menu.

1

Enter the type of task:
Medical
Enter the Integral Importance level of this task:
4
Enter the Deadline date of the task(FORMAT = DDMMYYYY):
31042024
Please Enter valid date in correct format.
31032024
Enter the Deadline time of the task(FORMAT = HHMMSS):
120000
Deadline already expired. Please Enter correct deadline.
Enter the Deadline date of the task(FORMAT = DDMMYYYY):
30042024
Enter the Deadline time of the task(FORMAT = HHMMSS):
120000
Enter the message you want with the remainder:
Go to Doctor
Task Added successfully.


Common-Task Menu:

Enter 1 to Insert a Task.
Enter 2 to delete a Task.
Enter 3 to delete all Common Tasks.
Enter 4 to View all the Common Tasks.
Enter 5 to Make some Edits in the Tasks.
Enter 6 to return to Main Menu.
```

Here now we inserted a common task. Insertion checks for valid deadline date and time and responds accordingly.Same way we 2 more tasks



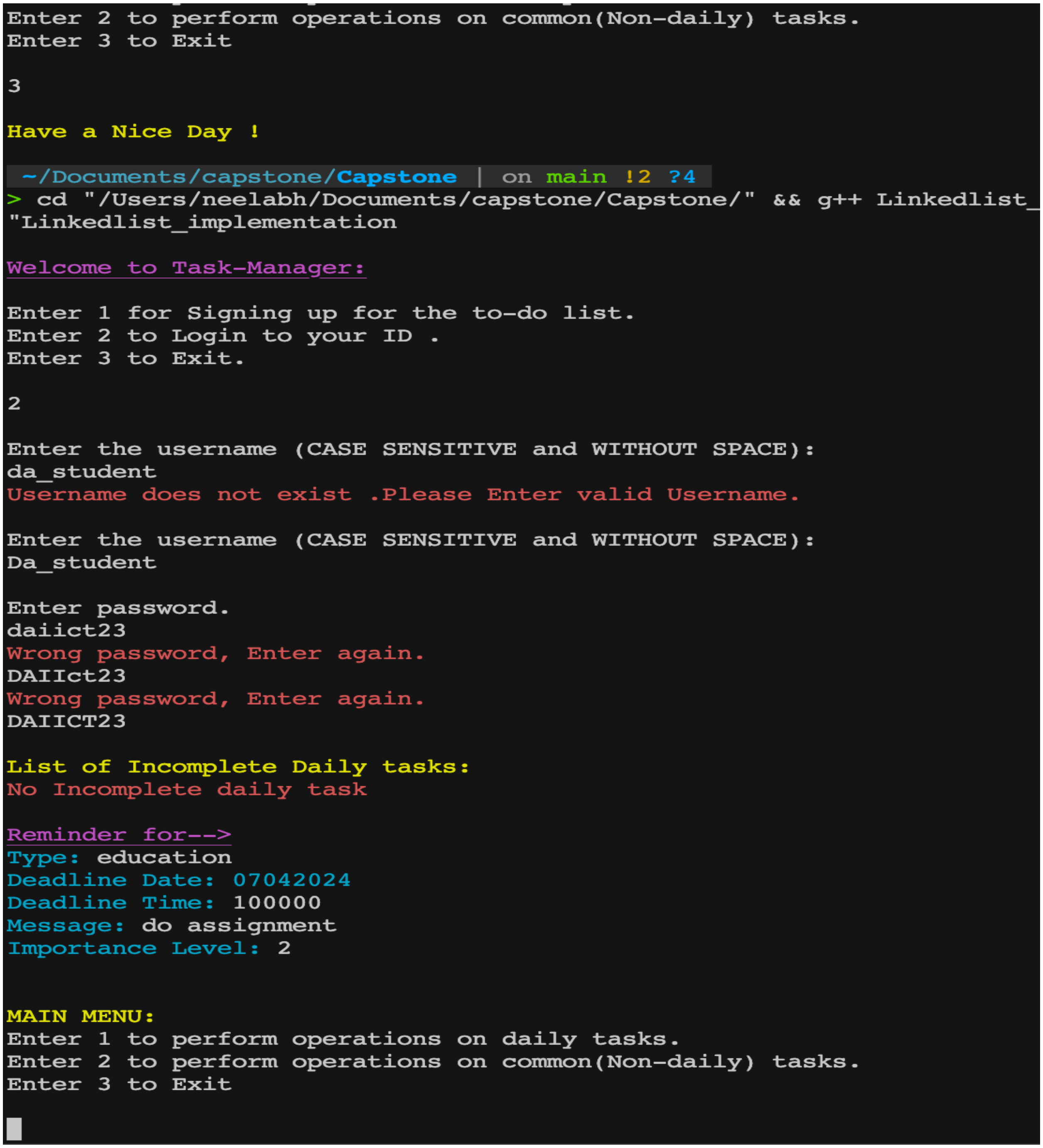
```
1

Enter the type of task:
Education
Enter the Integral Importance level of this task:
2
Enter the Deadline date of the task(FORMAT = DDMMYYYY):
07042024
Enter the Deadline time of the task(FORMAT = HHMMSS):
100000
Enter the message you want with the remainder:
do assignment
Task Added successfully.


Common-Task Menu:

Enter 1 to Insert a Task.
Enter 2 to delete a Task.
Enter 3 to delete all Common Tasks.
Enter 4 to View all the Common Tasks.
Enter 5 to Make some Edits in the Tasks.
Enter 6 to return to Main Menu.

1

Enter the type of task:
Personal
Enter the Integral Importance level of this task:
1
Enter the Deadline date of the task(FORMAT = DDMMYYYY):
25042025
Enter the Deadline time of the task(FORMAT = HHMMSS):
100000
Enter the message you want with the remainder:
Do something
Task Added successfully.


Common-Task Menu:

Enter 1 to Insert a Task.
Enter 2 to delete a Task.
Enter 3 to delete all Common Tasks.
Enter 4 to View all the Common Tasks.
Enter 5 to Make some Edits in the Tasks.
Enter 6 to return to Main Menu.
```

Now we return to main menu and exit the code and then login again



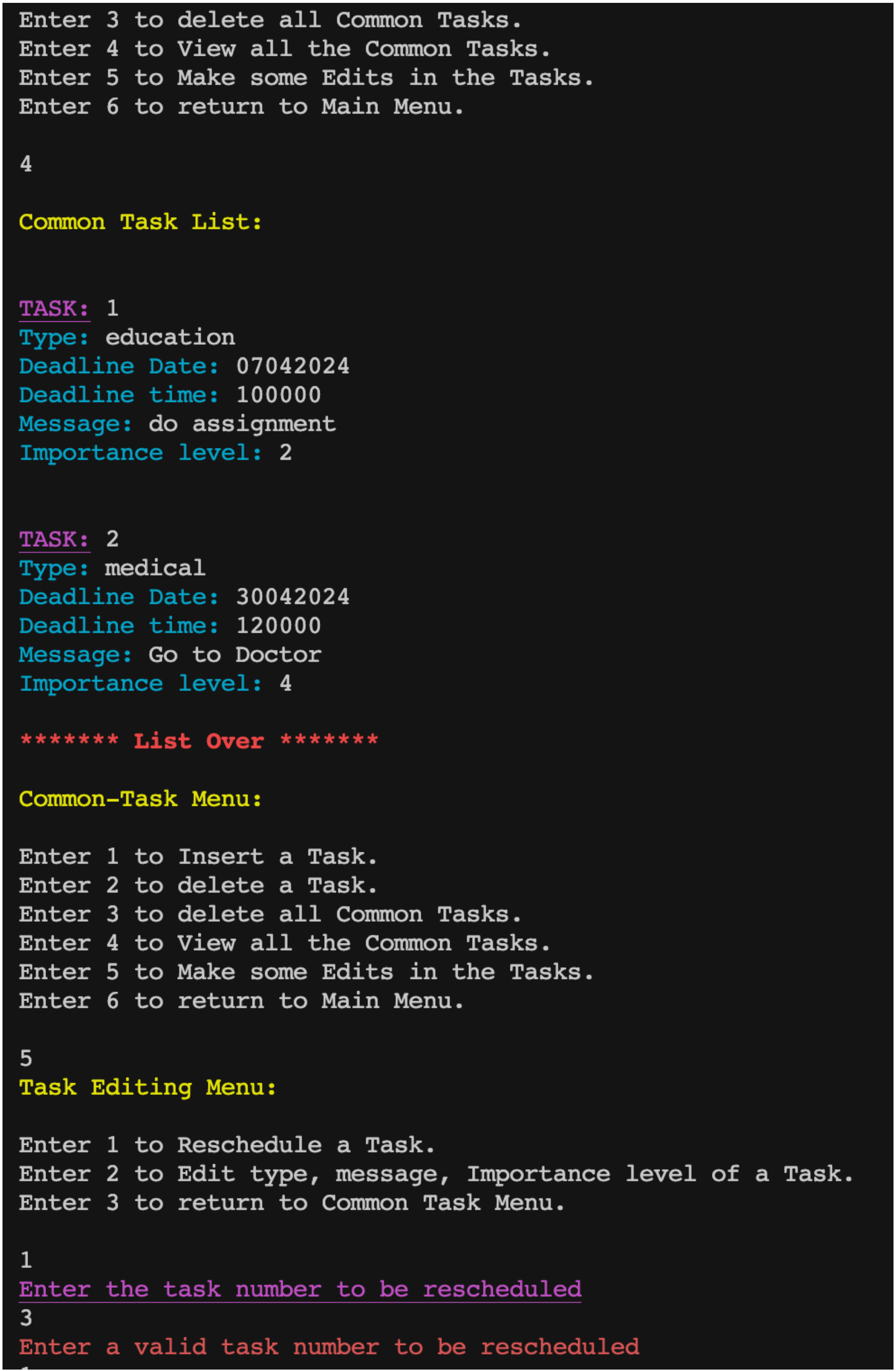
```
Enter 1 to Insert a Task.
Enter 2 to delete a Task.
Enter 3 to delete all Common Tasks.
Enter 4 to View all the Common Tasks.
Enter 5 to Make some Edits in the Tasks.
Enter 6 to return to Main Menu.

6

MAIN MENU:
Enter 1 to perform operations on daily tasks.
Enter 2 to perform operations on common(Non-daily) tasks.
Enter 3 to Exit

3

Have a Nice Day !
```

with the username and password.

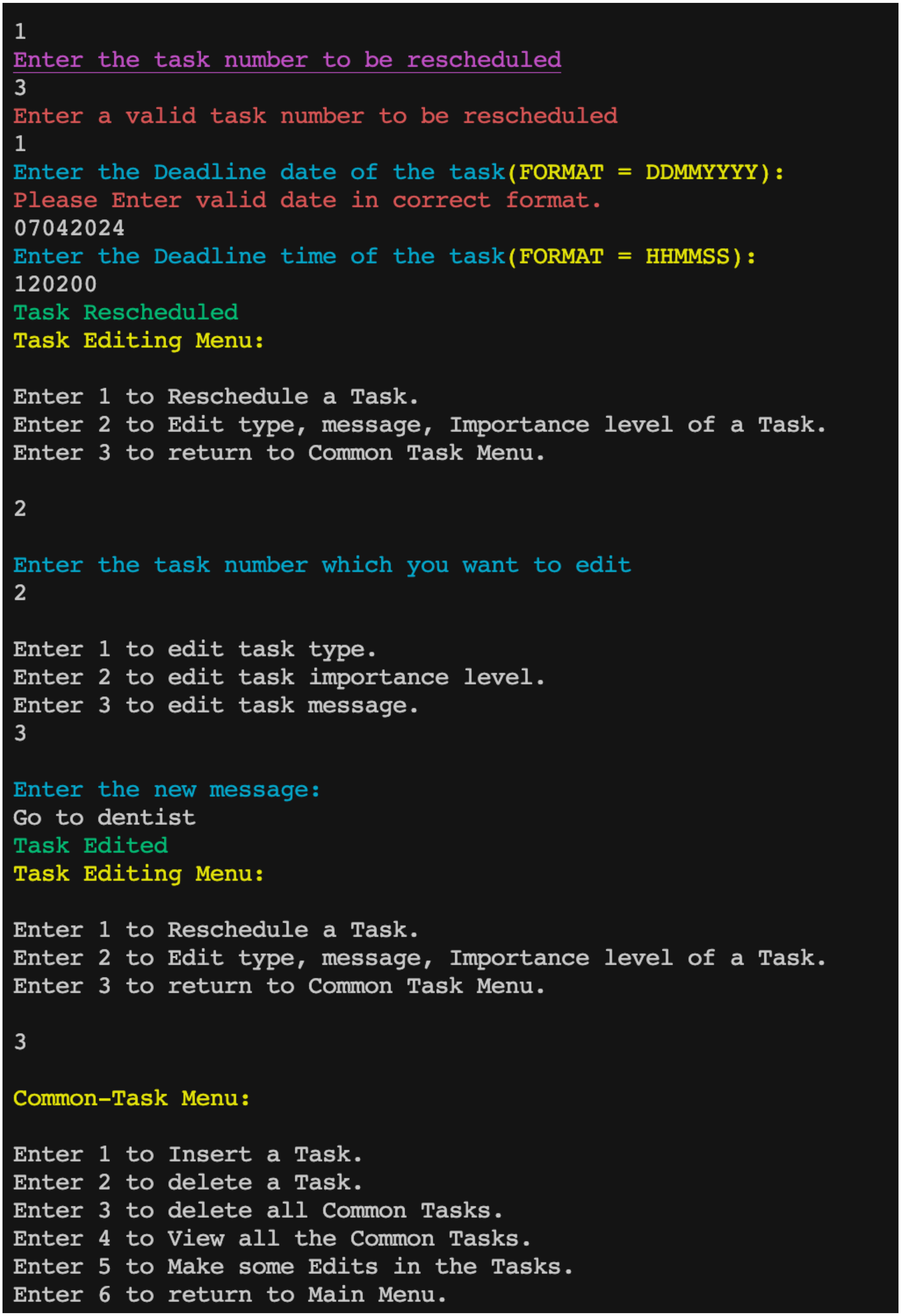


```
Enter 2 to perform operations on common(Non-daily) tasks.
Enter 3 to Exit

3

Have a Nice Day !

 ~/Documents/capstone/Capstone | on main !2 ?4
> cd "/Users/neelabh/Documents/capstone/Capstone/" && g++ Linkedlist_
"Linkedlist_implementation

Welcome to Task-Manager:

Enter 1 for Signing up for the to-do list.
Enter 2 to Login to your ID .
Enter 3 to Exit.

2

Enter the username (CASE SENSITIVE and WITHOUT SPACE):
da_student
Username does not exist .Please Enter valid Username.

Enter the username (CASE SENSITIVE and WITHOUT SPACE):
Da_student

Enter password.
daiict23
Wrong password, Enter again.
DAIIct23
Wrong password, Enter again.
DAIICT23

List of Incomplete Daily tasks:
No Incomplete daily task

Reminder for-->
Type: education
Deadline Date: 07042024
Deadline Time: 100000
Message: do assignment
Importance Level: 2


MAIN MENU:
Enter 1 to perform operations on daily tasks.
Enter 2 to perform operations on common(Non-daily) tasks.
Enter 3 to Exit

█
```

Username and password needs to be correct for logging in . As we login we are shown the List of Incomplete daily task. As per the time(11:51 PM) there was no incomplete task , so list is empty and Reminder for the common task which has less than 12 hr. Left for the expiry is shown. Then we deleted task 3 from the common task list using the delete function.

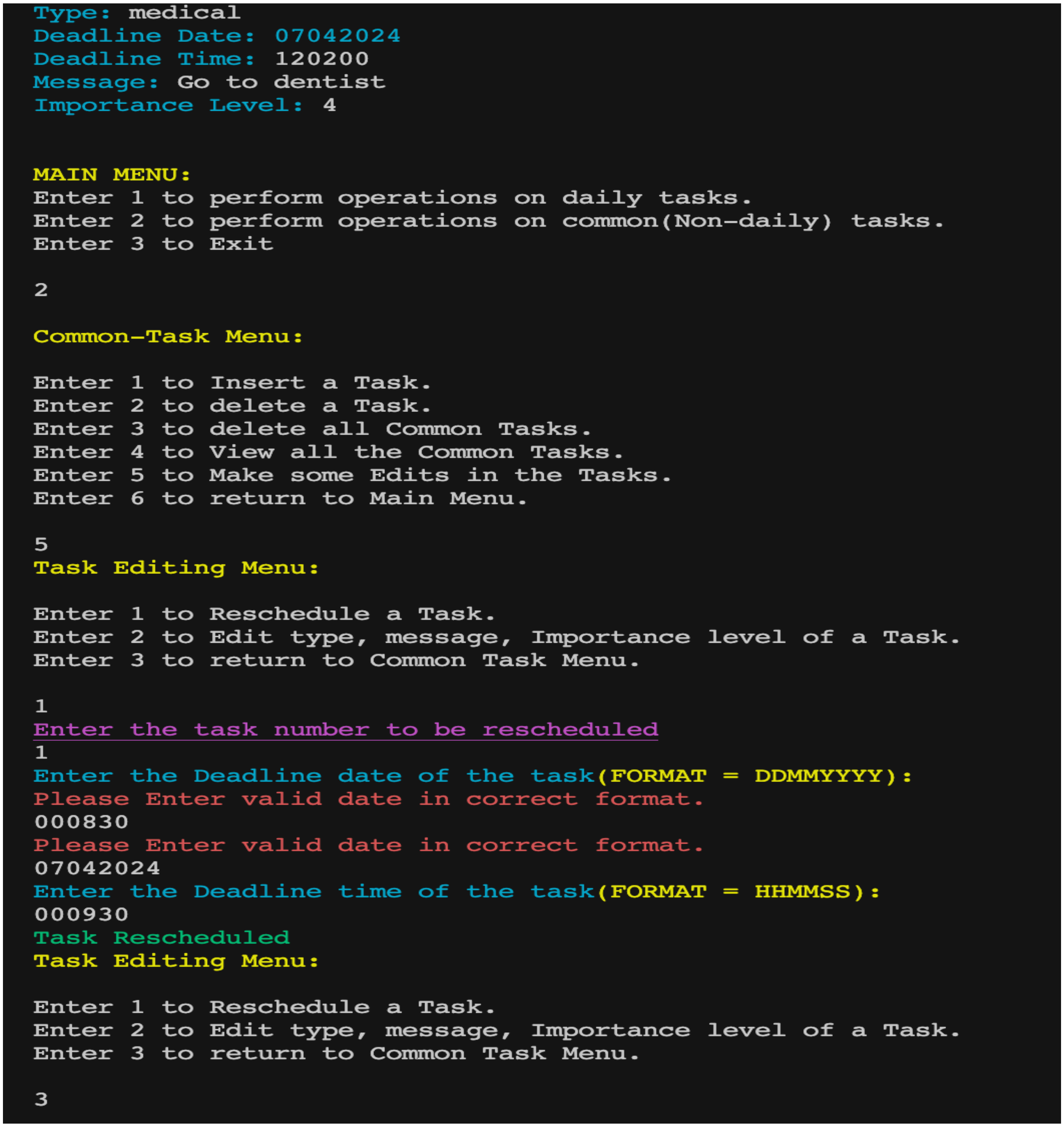


```
Enter 3 to delete all Common Tasks.
Enter 4 to View all the Common Tasks.
Enter 5 to Make some Edits in the Tasks.
Enter 6 to return to Main Menu.

4

Common Task List:


TASK: 1
Type: education
Deadline Date: 07042024
Deadline time: 100000
Message: do assignment
Importance level: 2


TASK: 2
Type: medical
Deadline Date: 30042024
Deadline time: 120000
Message: Go to Doctor
Importance level: 4

******* List Over *******

Common-Task Menu:

Enter 1 to Insert a Task.
Enter 2 to delete a Task.
Enter 3 to delete all Common Tasks.
Enter 4 to View all the Common Tasks.
Enter 5 to Make some Edits in the Tasks.
Enter 6 to return to Main Menu.

5
Task Editing Menu:

Enter 1 to Reschedule a Task.
Enter 2 to Edit type, message, Importance level of a Task.
Enter 3 to return to Common Task Menu.

1
Enter the task number to be rescheduled
3
Enter a valid task number to be rescheduled
```



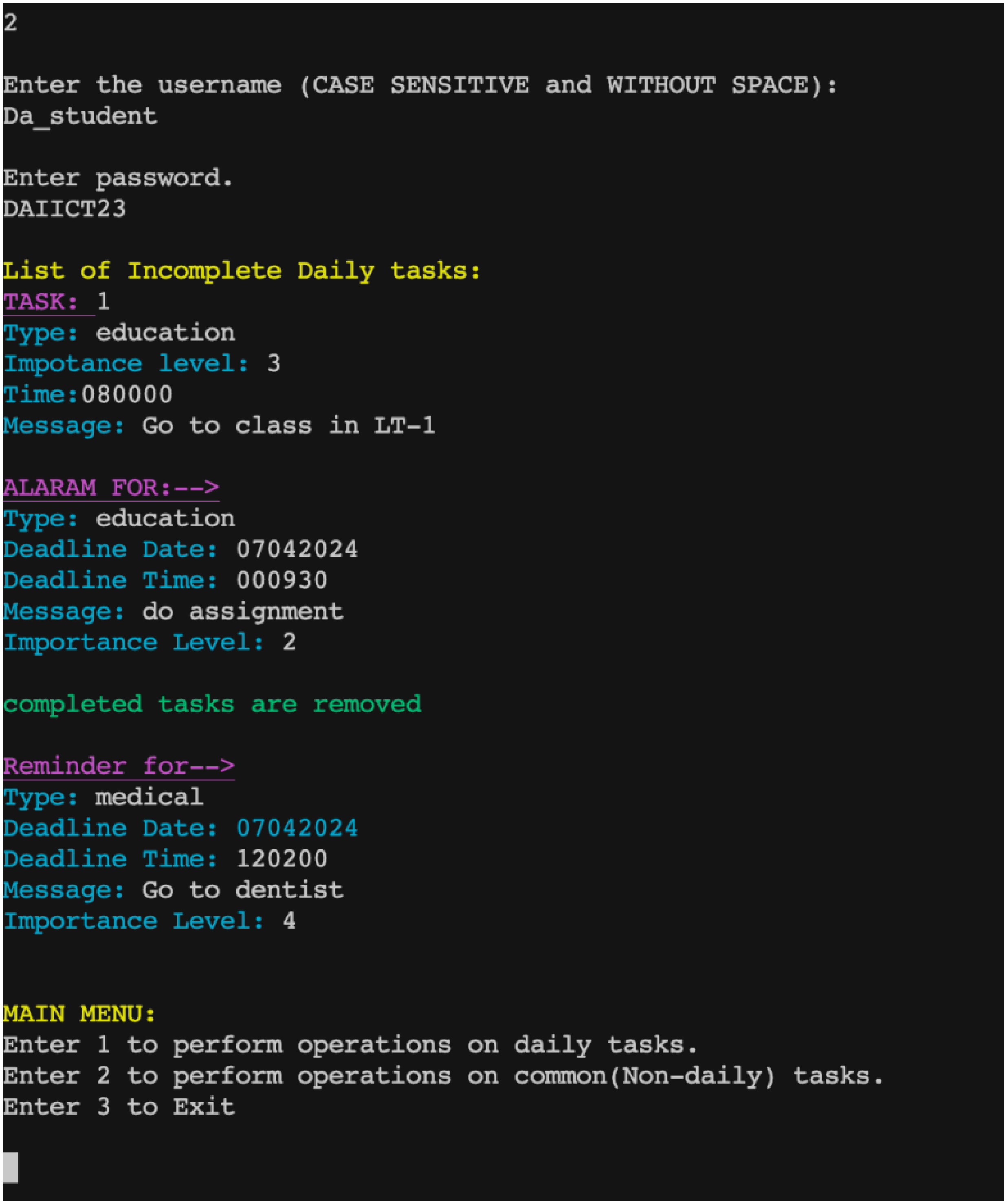
```
1
Enter the task number to be rescheduled
3
Enter a valid task number to be rescheduled
1
Enter the Deadline date of the task(FORMAT = DDMMYYYY):
Please Enter valid date in correct format.
07042024
Enter the Deadline time of the task(FORMAT = HHMMSS):
120200
Task Rescheduled
Task Editing Menu:

Enter 1 to Reschedule a Task.
Enter 2 to Edit type, message, Importance level of a Task.
Enter 3 to return to Common Task Menu.

2

Enter the task number which you want to edit
2

Enter 1 to edit task type.
Enter 2 to edit task importance level.
Enter 3 to edit task message.
3

Enter the new message:
Go to dentist
Task Edited
Task Editing Menu:

Enter 1 to Reschedule a Task.
Enter 2 to Edit type, message, Importance level of a Task.
Enter 3 to return to Common Task Menu.

3

Common-Task Menu:

Enter 1 to Insert a Task.
Enter 2 to delete a Task.
Enter 3 to delete all Common Tasks.
Enter 4 to View all the Common Tasks.
Enter 5 to Make some Edits in the Tasks.
Enter 6 to return to Main Menu.
```

We can view all the common task and then edit the tasks also. Here we edited the deadline time of task 1 and message of task 2.
Then we exit the code and again enter the code.
As we login again, (date was changed and time was 12:05 AM), so we got the incomplete daily task of 7 th april 2024 and reminder for both the common tasks as less then 12 hr. Was left for the expiry of their deadline.
Then we rescheduled the task1 of common list to such that when we re login , its deadline is over.



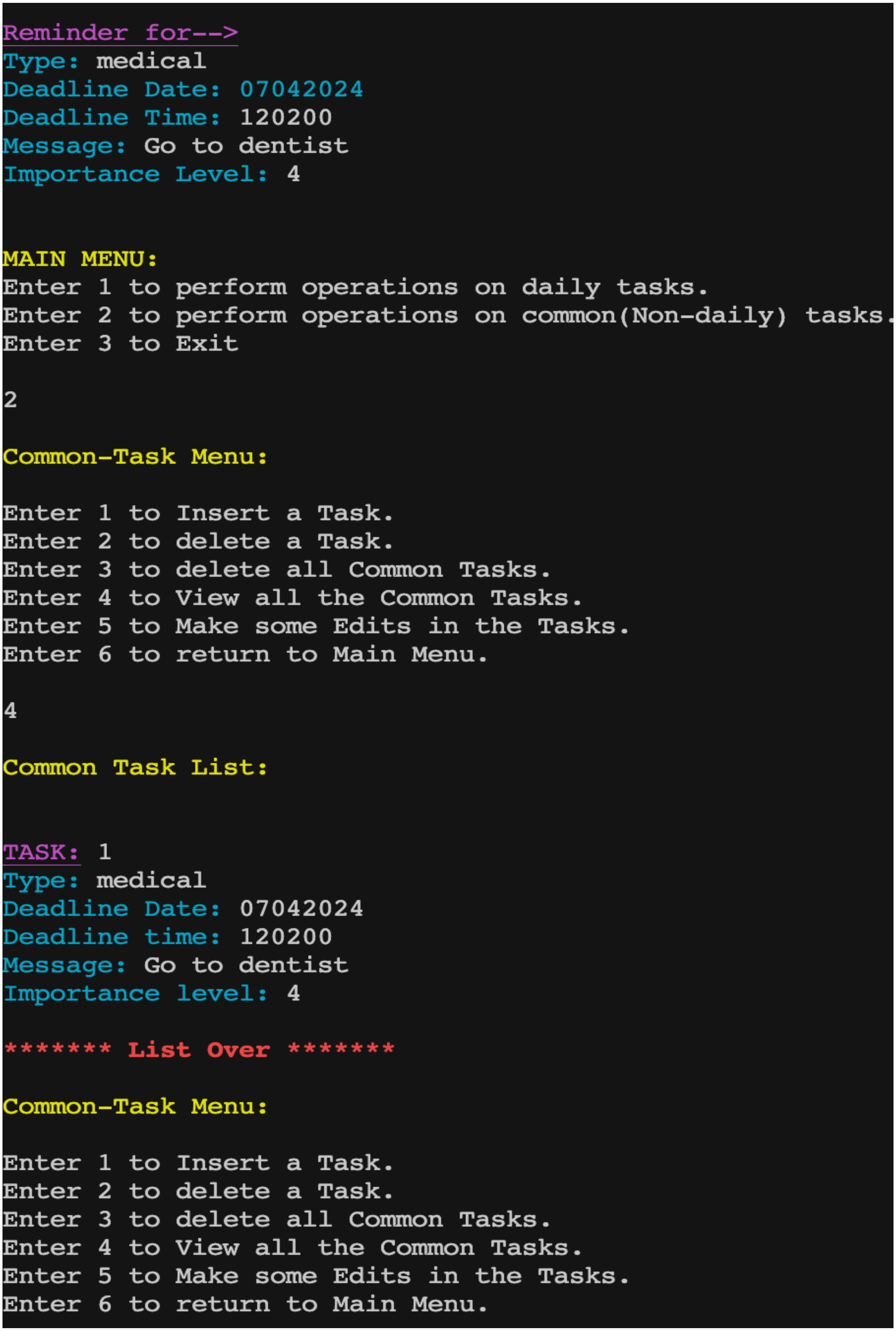
```
Type: medical
Deadline Date: 07042024
Deadline Time: 120200
Message: Go to dentist
Importance Level: 4


MAIN MENU:
Enter 1 to perform operations on daily tasks.
Enter 2 to perform operations on common(Non-daily) tasks.
Enter 3 to Exit

2

Common-Task Menu:

Enter 1 to Insert a Task.
Enter 2 to delete a Task.
Enter 3 to delete all Common Tasks.
Enter 4 to View all the Common Tasks.
Enter 5 to Make some Edits in the Tasks.
Enter 6 to return to Main Menu.

5
Task Editing Menu:

Enter 1 to Reschedule a Task.
Enter 2 to Edit type, message, Importance level of a Task.
Enter 3 to return to Common Task Menu.

1
Enter the task number to be rescheduled
1
Enter the Deadline date of the task(FORMAT = DDMMYYYY):
Please Enter valid date in correct format.
000830
Please Enter valid date in correct format.
07042024
Enter the Deadline time of the task(FORMAT = HHMMSS):
000930
Task Rescheduled
Task Editing Menu:

Enter 1 to Reschedule a Task.
Enter 2 to Edit type, message, Importance level of a Task.
Enter 3 to return to Common Task Menu.

3
```

Now when we login again we will get one completed task and it will be removed from the list.



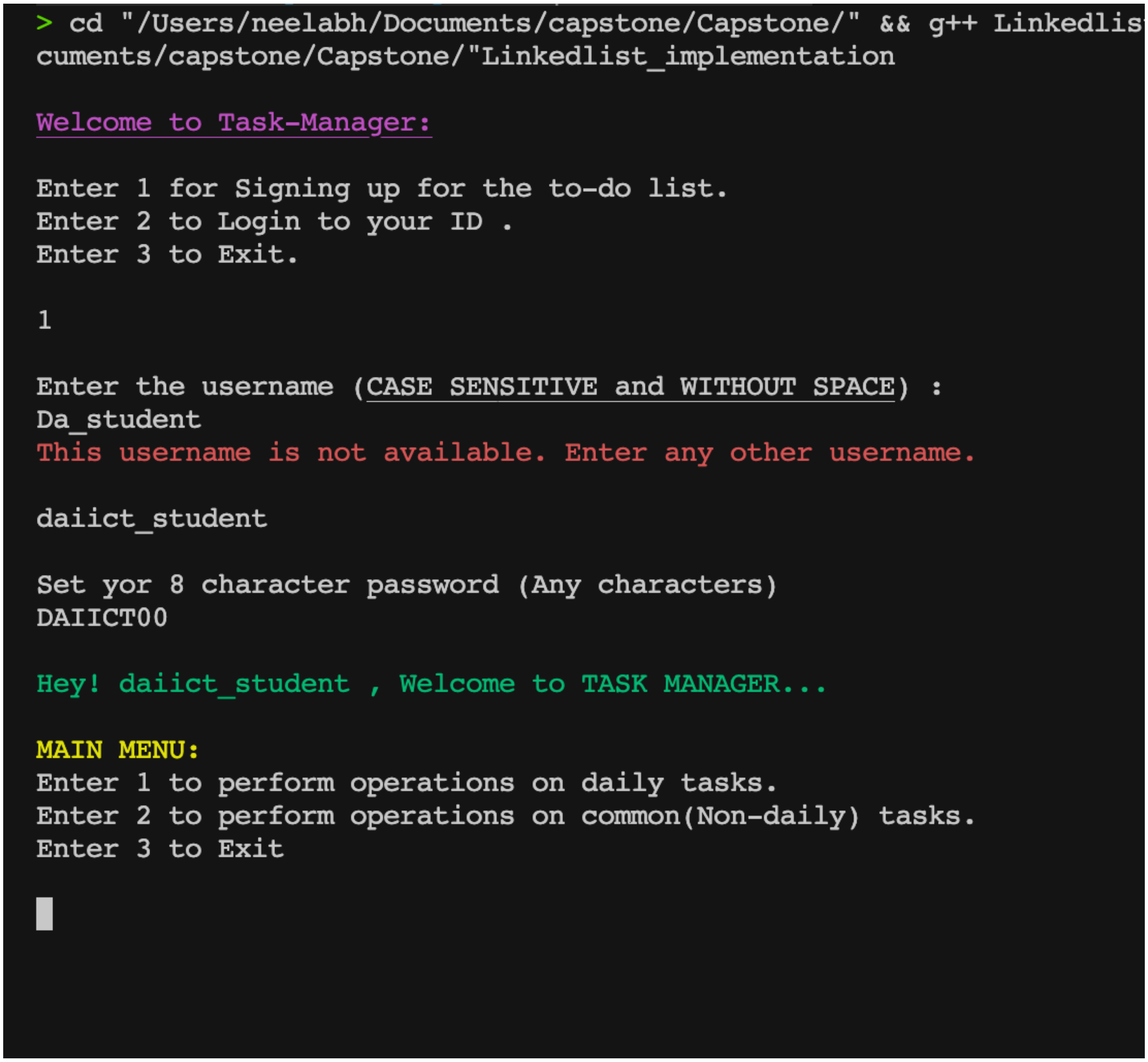
```
2

Enter the username (CASE SENSITIVE and WITHOUT SPACE):
Da_student

Enter password.
DAIICT23

List of Incomplete Daily tasks:
TASK: 1
Type: education
Impotance level: 3
Time:080000
Message: Go to class in LT-1

ALARAM FOR:-->
Type: education
Deadline Date: 07042024
Deadline Time: 000930
Message: do assignment
Importance Level: 2

completed tasks are removed

Reminder for-->
Type: medical
Deadline Date: 07042024
Deadline Time: 120200
Message: Go to dentist
Importance Level: 4


MAIN MENU:
Enter 1 to perform operations on daily tasks.
Enter 2 to perform operations on common(Non-daily) tasks.
Enter 3 to Exit
```

Here we got alarm for the completed task



```
Reminder for-->
Type: medical
Deadline Date: 07042024
Deadline Time: 120200
Message: Go to dentist
Importance Level: 4


MAIN MENU:
Enter 1 to perform operations on daily tasks.
Enter 2 to perform operations on common(Non-daily) tasks.
Enter 3 to Exit

2

Common-Task Menu:

Enter 1 to Insert a Task.
Enter 2 to delete a Task.
Enter 3 to delete all Common Tasks.
Enter 4 to View all the Common Tasks.
Enter 5 to Make some Edits in the Tasks.
Enter 6 to return to Main Menu.

4

Common Task List:


TASK: 1
Type: medical
Deadline Date: 07042024
Deadline time: 120200
Message: Go to dentist
Importance level: 4

******* List Over *******

Common-Task Menu:

Enter 1 to Insert a Task.
Enter 2 to delete a Task.
Enter 3 to delete all Common Tasks.
Enter 4 to View all the Common Tasks.
Enter 5 to Make some Edits in the Tasks.
Enter 6 to return to Main Menu.
```

And that task is removed from the common list.
Also we can erase all the common task list using the function.
Now if we try to sign up using the already used username we will get error.



```
> cd "/Users/neelabh/Documents/capstone/Capstone/" && g++ Linkedlis
cuments/capstone/Capstone/"Linkedlist_implementation

Welcome to Task-Manager:

Enter 1 for Signing up for the to-do list.
Enter 2 to Login to your ID .
Enter 3 to Exit.

1

Enter the username (CASE SENSITIVE and WITHOUT SPACE) :
Da_student
This username is not available. Enter any other username.

daiict_student

Set yor 8 character password (Any characters)
DAIICT00

Hey! daiict_student , Welcome to TASK MANAGER...

MAIN MENU:
Enter 1 to perform operations on daily tasks.
Enter 2 to perform operations on common(Non-daily) tasks.
Enter 3 to Exit

```

Here , we can see that the errors are handled and a new sign up form username : daiict_student is done. We now have a completely different user-id from the one of Da_student. So all the tasks of daiict_student will be stored separately from that of da_student. Using this we can have multiple users, with each having privacy through their password.

Time and Space Complexities for implementation using linked list

1 string To Int(str)

Time Complexity: O(n)
Space Complexity: O(1)

2 extract Date Components(date, day, month, year)

Time Complexity: O(1)
Space Complexity: O(1)

3 extract Time Components(time, hours, minutes, seconds)

Time Complexity: O(1)
Space Complexity: O(1)

4 correct date(deadline date)

Time Complexity: O(1)
Space Complexity: O(1)

5 correct time(deadline time)

Time Complexity: O(1)
Space Complexity: O(1)

6 priority(temp, temp insert)

Time Complexity: O(1)
Space Complexity: O(1)

7 insert task common(deadline date, deadline time, type by user, message, imp lvl by user)

Time Complexity: O(n)
Space Complexity: O(1)

8 compare time ctime user time(user date, user time)

Time Complexity: O(1)
Space Complexity: O(1)

9 insert task()

Time Complexity: O(n)
Space Complexity: O(1)

10 print common list(head)

Time Complexity: O(n)
Space Complexity: O(1)

11 write Data To File(username)

Time Complexity: O(n)
Space Complexity: O(1)

12 load Data From File(username)

Time Complexity: O(n)
Space Complexity: O(1)

13 erase File Contents(filename)

Time Complexity: O(1)
Space Complexity: O(1)

14 compare 2 time(current, c hour, c minute, c second, c date, c month, c year)

Time Complexity: O(1)
Space Complexity: O(1)

15 check date(month, date, year)

Time Complexity: O(1)
Space Complexity: O(1)

16 is list empty()

Time Complexity: O(1)
Space Complexity: O(1)

17 reminder x hr(r hour)

Time Complexity: O(n)
Space Complexity: O(1)

18 remove tasks(username)

Time Complexity: O(n)
Space Complexity: O(1)

19 alaram(head, username)

Time Complexity: O(n)
Space Complexity: O(1)

20 reschedule(username)

Time Complexity: O(n)
Space Complexity: O(1)

21 edit task(username)

Time Complexity: O(1)
Space Complexity: O(1)

22 compare time daily(deadline)

Time Complexity: O(1)
Space Complexity: O(1)

23 insert daily(deadline time, task type user, message, imp lvl user)

Time Complexity: O(n)
Space Complexity: O(1)

24 insert daily task()

Time Complexity: O(n)
Space Complexity: O(1)

25 display daily()

Time Complexity: O(n)
Space Complexity: O(1)

26 display incomplete dailytask()

Time Complexity: O(n)
Space Complexity: O(1)

27 delete daily task()

Time Complexity: O(n)
Space Complexity: O(1)

28 write daily task(username)

Time Complexity: O(n)
Space Complexity: O(1)

29 load daily task file(username)

Time Complexity: O(n)
Space Complexity: O(1)

30 erase all daily task(username)

Time Complexity: O(1)
Space Complexity: O(1)

31 loaduname(unames)

Time Complexity: O(n)
Space Complexity: O(1)

32 writeuname(unames)

Time Complexity: O(n)
Space Complexity: O(1)

33 writeuser pass(user pass)

Time Complexity: O(n)
Space Complexity: O(1)

34 readuser pass(user pass)

Time Complexity: O(n)
Space Complexity: O(1)

2.) Implementation using hash tables

- **We have successfully implemented the problem using linked list and it is amazing in performance , but the only problem is that it has most of its functions in time complexity O(n). To solve this issue and to improve time complexity we have made another implementation using hash tables which uses files to store the data. It is much more powerful in terms of time , but it is a light application and does not have as good user interface as compared to the linked list version.**

Pseudocode and time complexity for Implementation Using Tables

Date Class

- Declare private variables for day, month, and year
- Define constructor for Date class
  - Time complexity: $O(1)$
  - Space complexity: $O(1)$
- Define `ShowDateStr` method to return date as a string
  - Time complexity: $O(1)$
  - Space complexity: $O(1)$
- Define `ShowDateInt` method to return date as an integer for sorting
  - Time complexity: $O(1)$
  - Space complexity: $O(1)$
- Define friend function `Compare` for comparing dates and times
  - Time complexity: $O(1)$
  - Space complexity: $O(1)$

Time Class

- Declare private variables for hour, minute, and second
- Define constructor for Time class
  - Time complexity: $O(1)$
  - Space complexity: $O(1)$
- Define `ShowTimeInt` method to return time as an integer for sorting
  - Time complexity: $O(1)$
  - Space complexity: $O(1)$
- Define `ShowTimeStr` method to return time as a string
  - Time complexity: $O(1)$
  - Space complexity: $O(1)$
- Define friend function `Compare` for comparing dates and times
  - Time complexity: $O(1)$
  - Space complexity: $O(1)$

Task Class

- Declare private variables for priority, task description, linked status, date, time, task completion status, and execution status
- Define constructors for Task class
  - Time complexity: $O(1)$
  - Space complexity: $O(1)$
- Define friend functions `CreateTask`, `ShowTask`, and `deletetask`
  - Time complexity: $O(1)$
  - Space complexity: $O(1)$

Head Class

- Declare private variable for size of linked list
- Declare pointers for Front and Rear of linked list
- Define friend functions `Push`, `Show`, and `PopFront` for managing linked list
  - Time complexity: $O(1)$ for `Push`, $O(n)$ for `Show`, $O(1)$ for `PopFront`
  - Space complexity: $O(1)$

Node Class

- Declare private variable for storing a task
- Declare pointer for the next node in linked list
- Define friend functions `Push`, `Show`, and `PopFront` for managing linked list
  - Time complexity: $O(1)$
  - Space complexity: $O(1)$

Push Function

- Allocate memory for a new node
- Handle the case when the list is empty
- Traverse the list to find the correct position to insert the task based on its priority
- Insert the task into the linked list at the correct position
  - Time complexity: $O(n)$ in the worst case
  - Space complexity: $O(1)$

Show Function

- Retrieve tasks from the linked list
- Display tasks or perform operations on them
  - Time complexity: $O(1)$
  - Space complexity: $O(1)$

PopFront Function

- Remove the first node from the linked list
- Update pointers accordingly
  - Time complexity: $O(n)$
  - Space complexity: $O(1)$

CreateTask Function

- Get input for task details from user
- Create a task object and store it in a file
  - Time complexity: $O(1)$
  - Space complexity: $O(1)$

ShowTask Function

- Check for tasks scheduled for the current time
- Display or perform operations on the tasks
  - Time complexity: $O(1)$
  - Space complexity: $O(1)$

deletetask Function

- Get input for date and time of the task to be deleted
- Delete the corresponding file if it exists
  - Time complexity: $O(1)$
  - Space complexity: $O(1)$

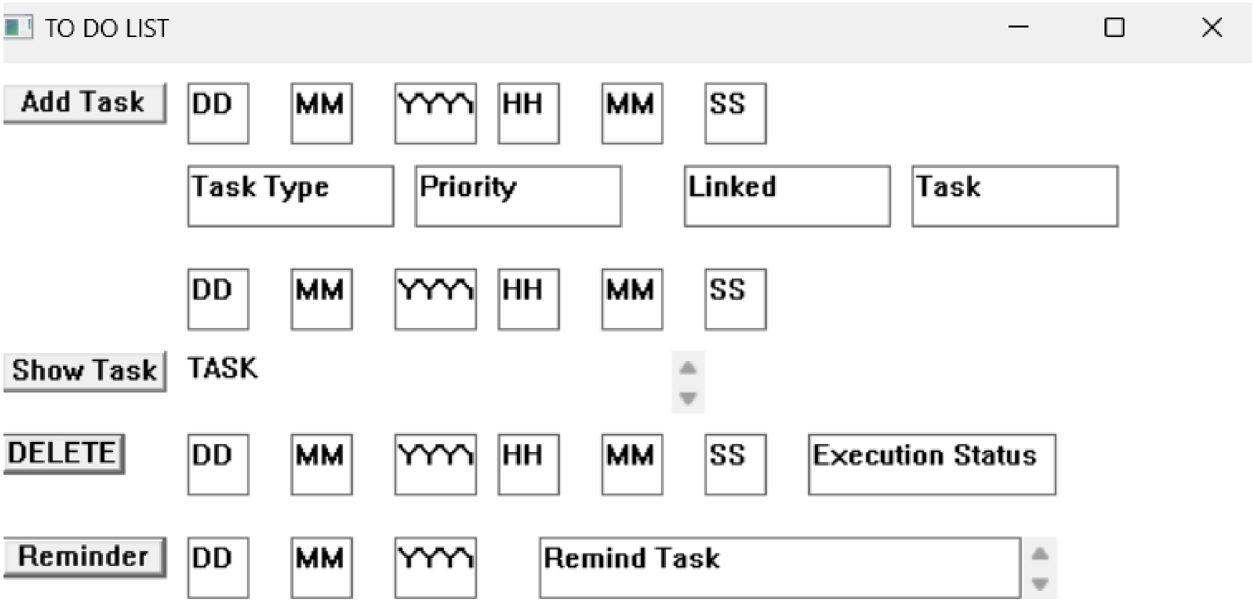
UserInterface Function

- Display menu options to the user
- Handle user input and call corresponding functions
  - Time complexity: $O(1)$
  - Space complexity: $O(1)$

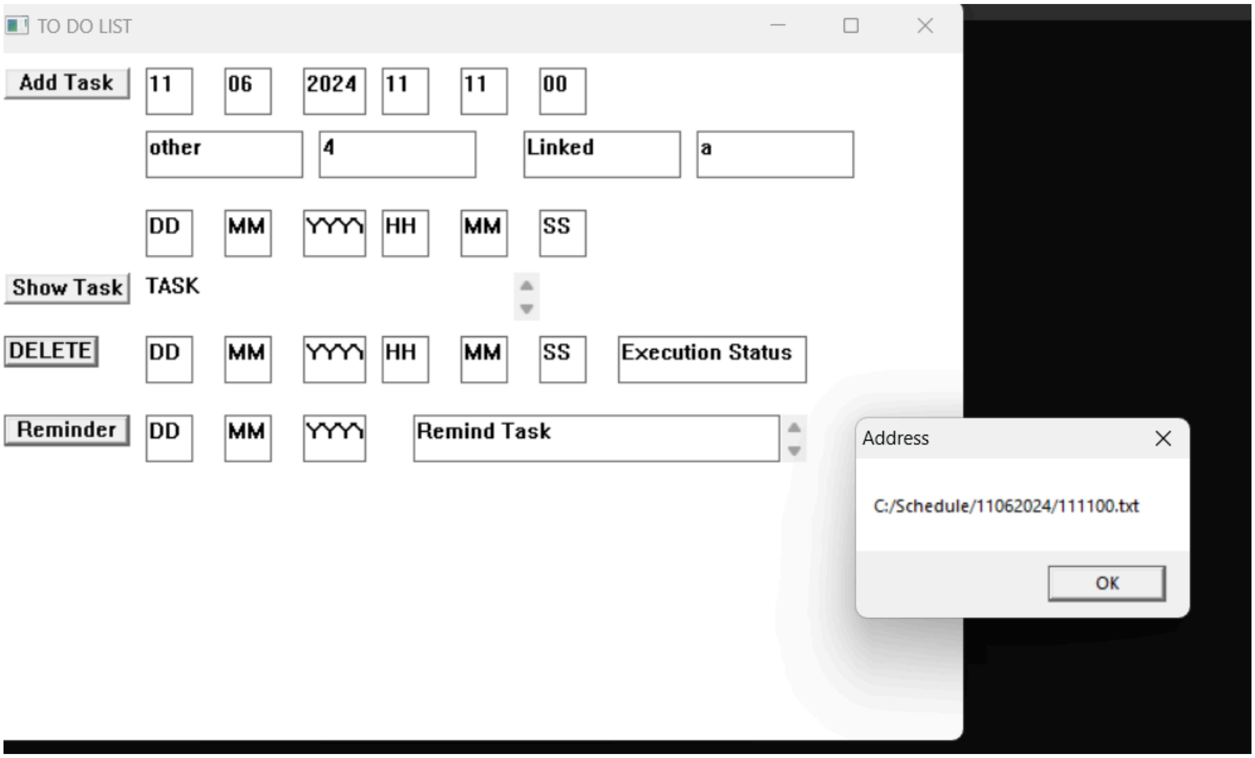
Main Function

- Call the `UserInterface` function to start the program
  - Time complexity: $O(1)$
  - Space complexity: $O(1)$

GUI FOR IMPLEMENTATION USING TABLES



The following is the application made with c++, to implement the tasks delete them show the tasks, and get a reminder. The new thing in this implementation is the linked function which sees that is that linked with previous task or not. It would be 0 or 1 depending on that.

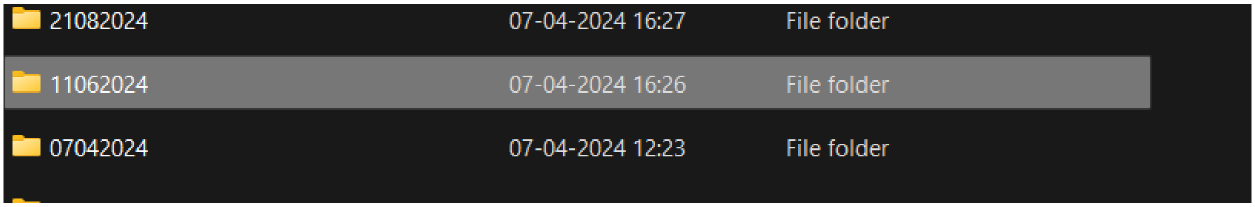


11 06 2024 11 11 00

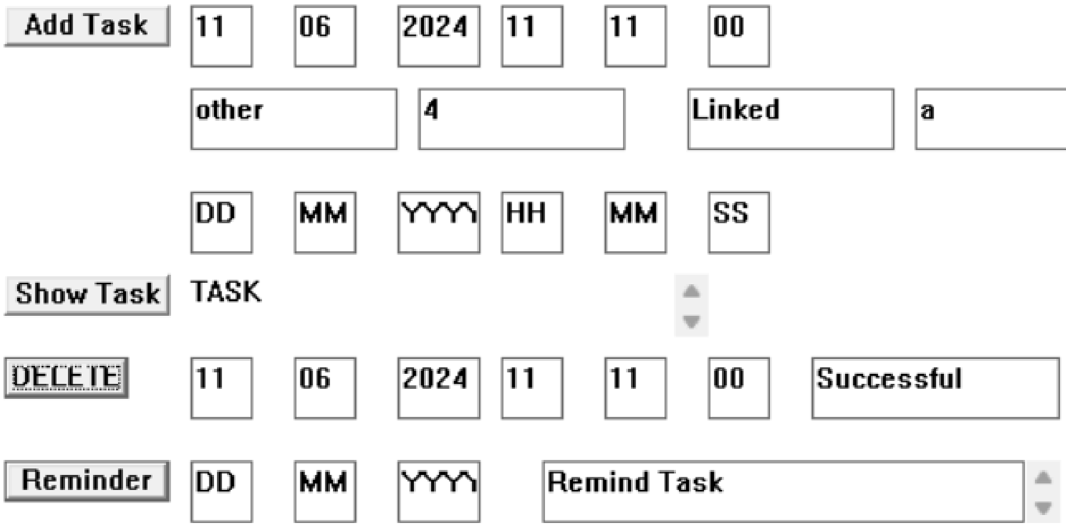
other 4 Linked a

C:/Schedule/11062024/111100.txt

The task is added and the file is saved in the computer.



21082024	07-04-2024 16:27	File folder
11062024	07-04-2024 16:26	File folder
07042024	07-04-2024 12:23	File folder



Add Task 11 06 2024 11 11 00

other 4 Linked a

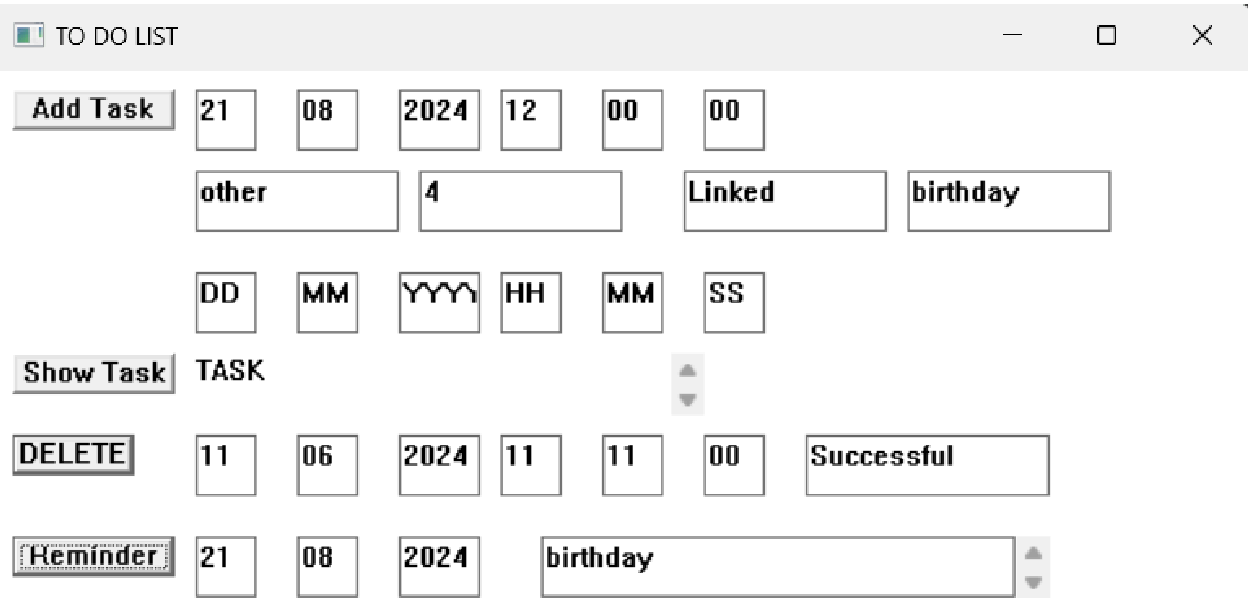
DD MM YYYY HH MM SS

Show Task TASK

DELETE 11 06 2024 11 11 00 Successful

Reminder DD MM YYYY Remind Task

The delete function deletes the folder of the date which is written in the box. If the date written in the box is not in the memory of the computer , then it throws an error message.



21 08 2024 12 00 00

other 4 Linked birthday

DD MM YYYY HH MM SS

TASK

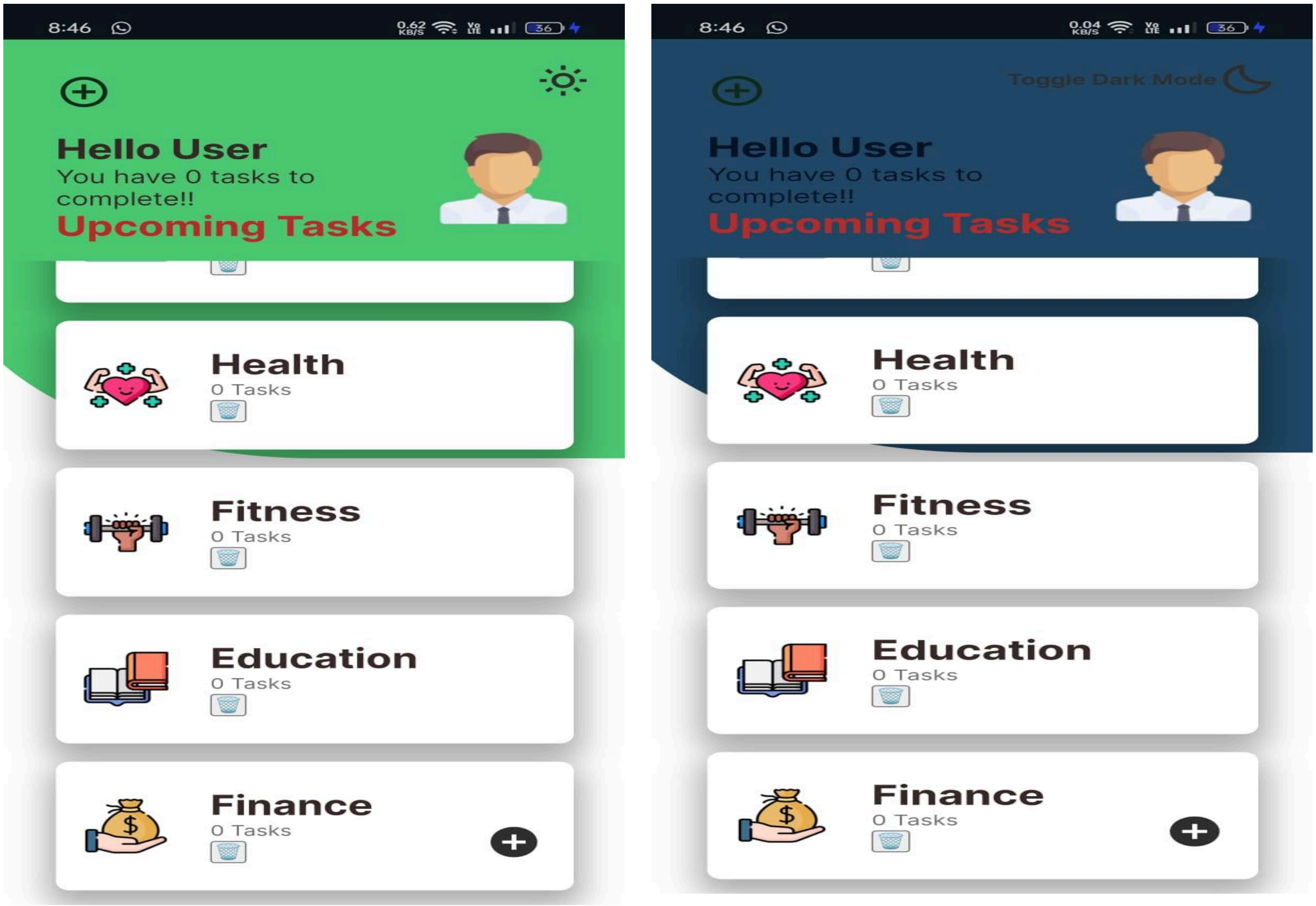
11 06 2024 11 11 00 Successful

21 08 2024 birthday

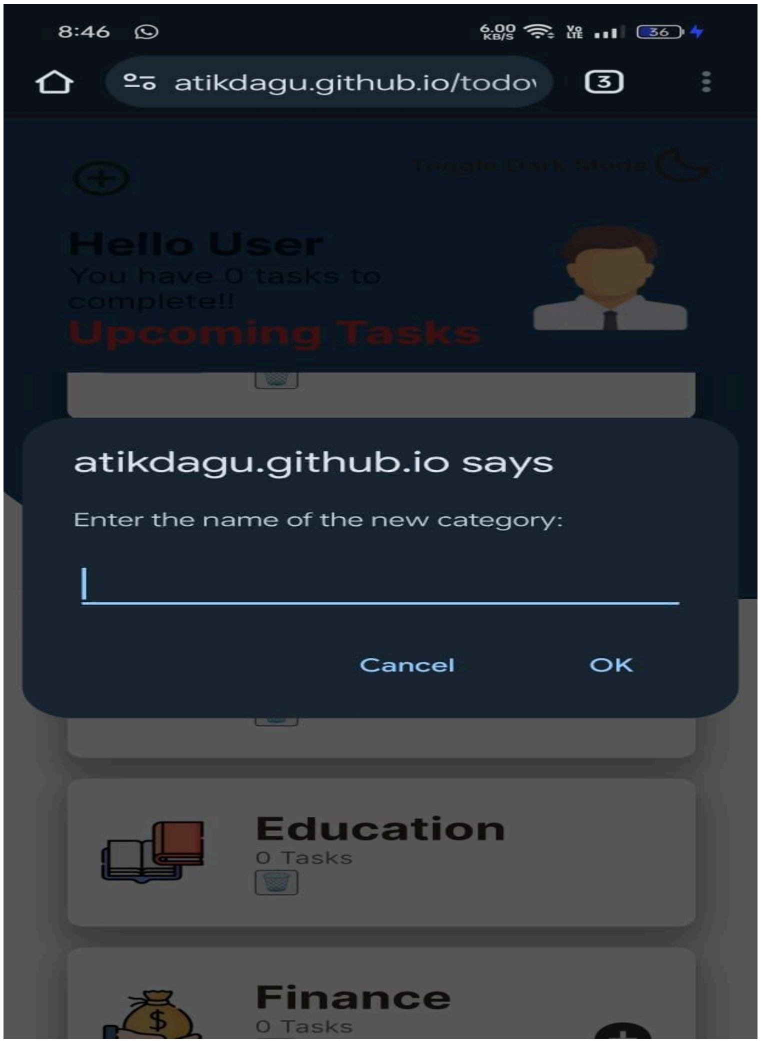
The reminder function takes the date of the task and prints the task in the given folder.

GUI USING WEBSITE

Website_link:-https://atikdagu.github.io/todoweb/



There are two modes in the website dark and light



Hello User

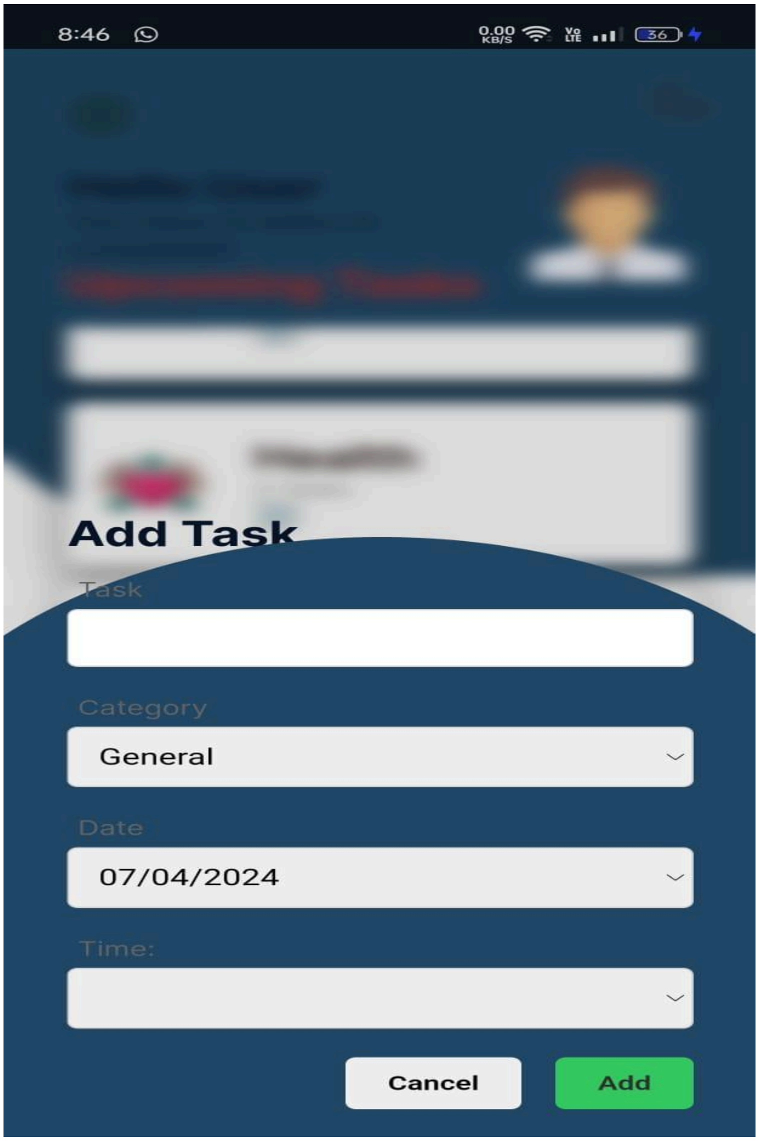
You have 0 tasks to complete!!

Upcoming Tasks

atikdagu.github.io says

Enter the name of the new category:

Category section



Add Task

Task

Category

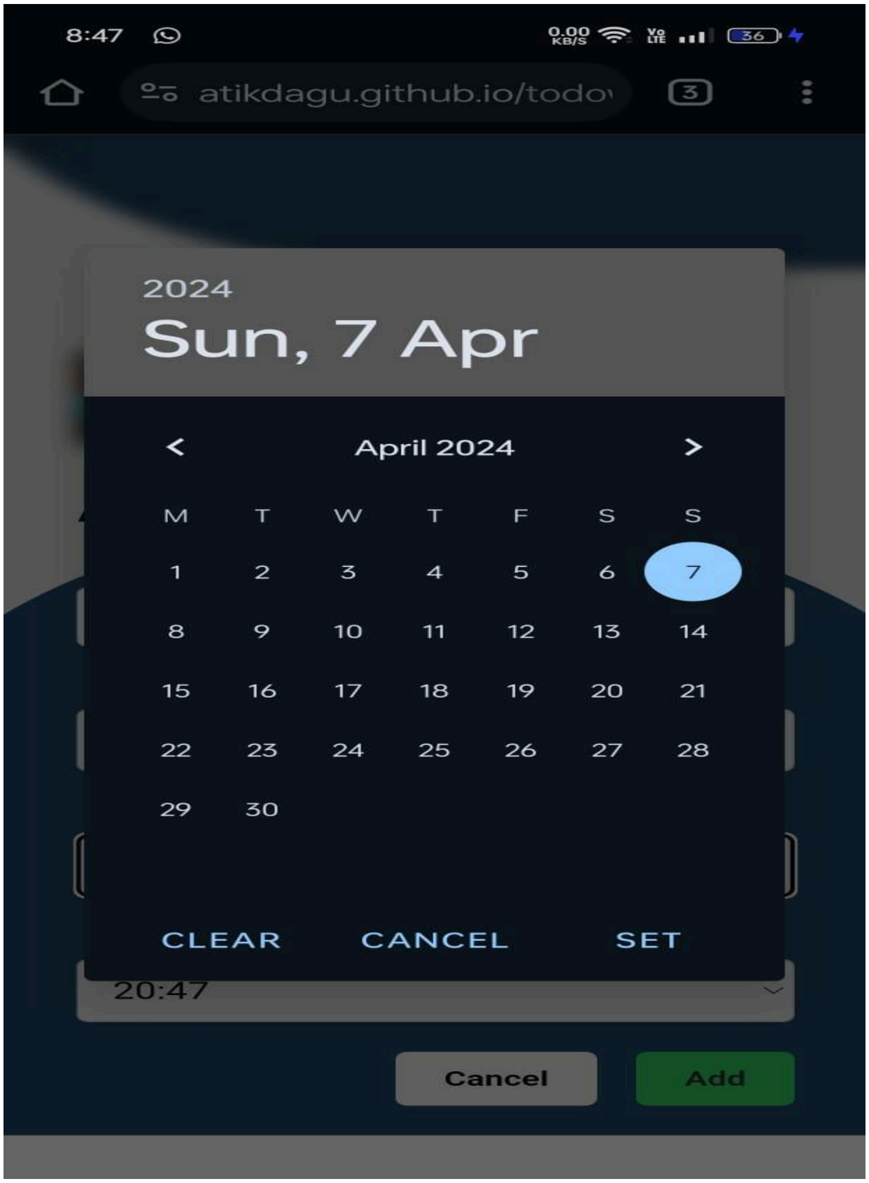
General

Date

07/04/2024

Time:

Add task section



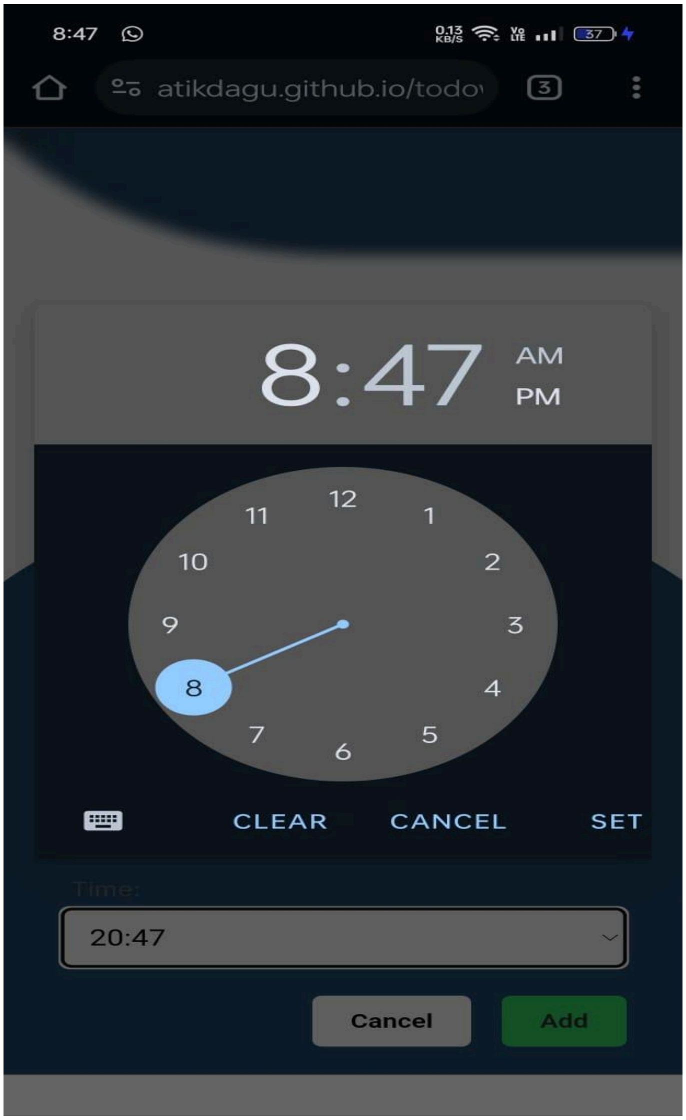
2024

Sun, 7 Apr

M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

20:47

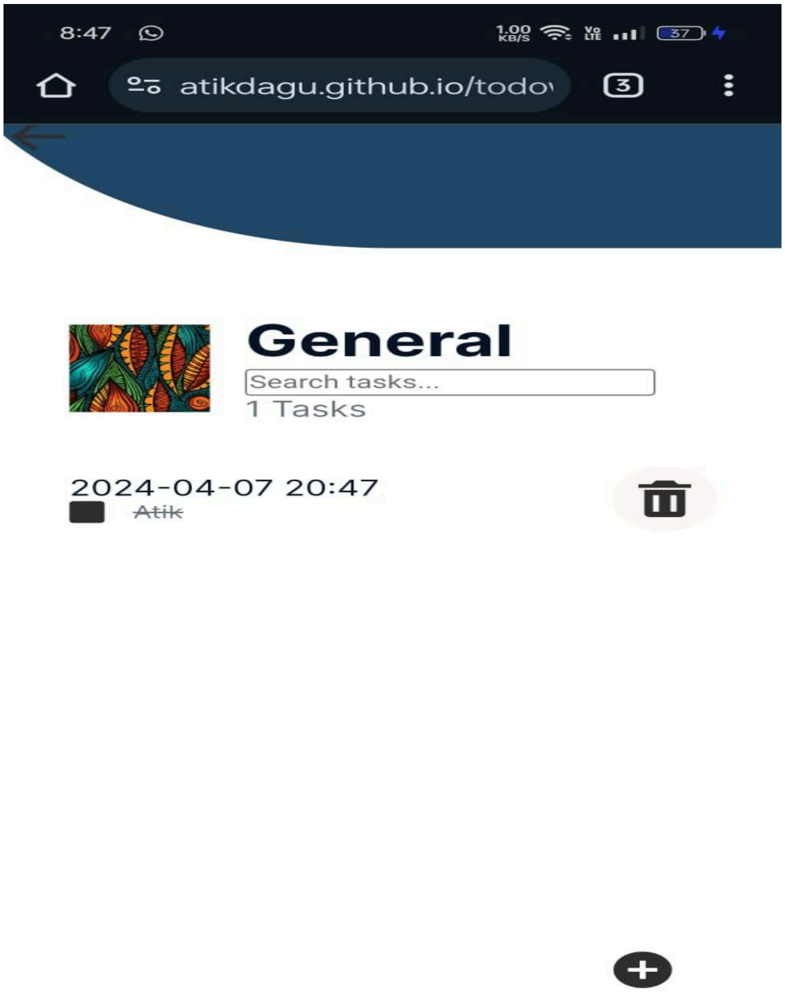
Date section



Time:

20:47

Time section



General

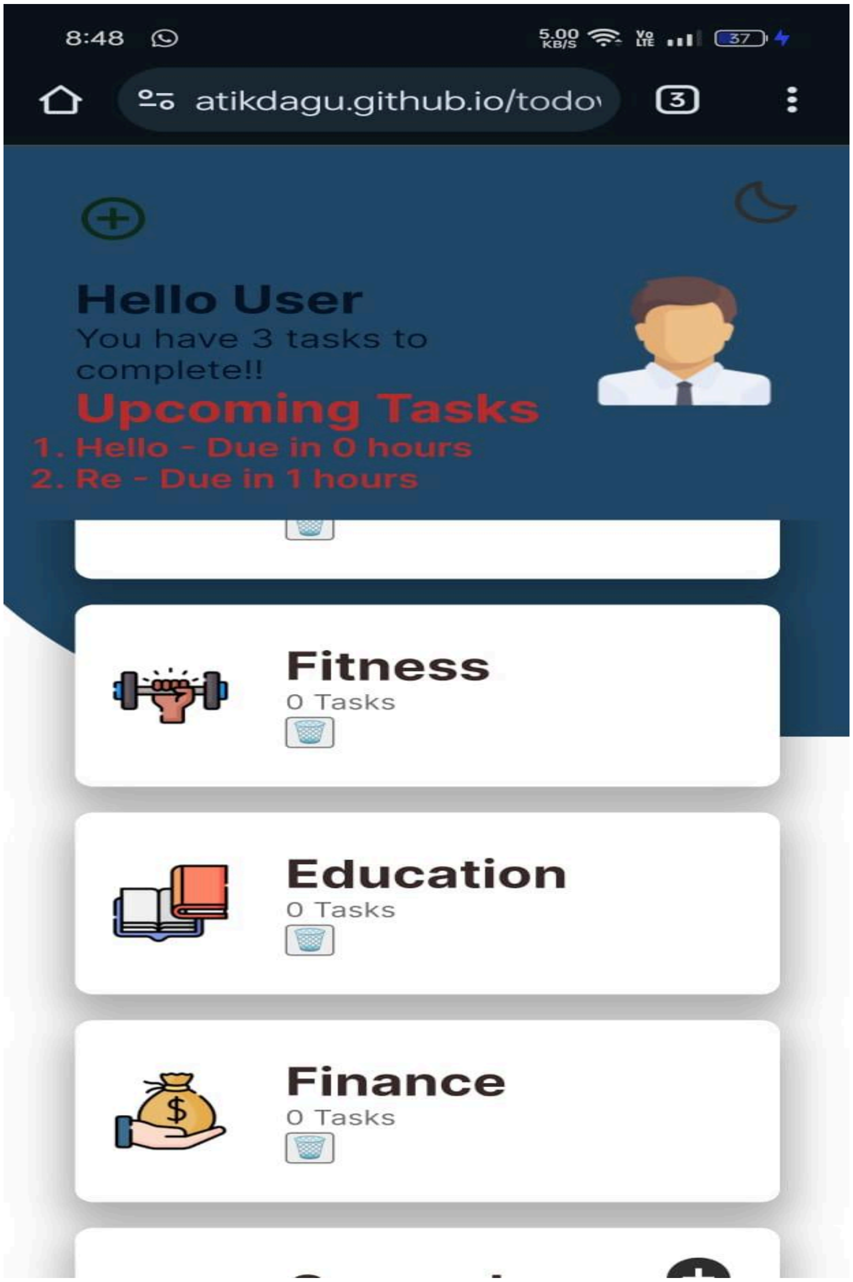
Search tasks...

1 Tasks

2024-04-07 20:47

■ ~~Atik~~

Checkbox



8:48

atikdagu.github.io/todo

Hello User

You have 3 tasks to complete!!

Upcoming Tasks

1. Hello - Due in 0 hours
2. Re - Due in 1 hours

Fitness

0 Tasks

Education

0 Tasks

Finance

0 Tasks

Showing upcoming tasks