UNIT TESTING

Below are a series of unit testing outputs for a task management system.

Test Case Script Instructions:

Ensure your Python installation has Pytest.

Set your Python terminal to the directory of our code repository.

Enter in the Python terminal the command ‘pytest’, which will run test scripts and generate results in the Python terminal.

User Registration:

- *Users will be able to register for an account with a username, password and will be given a priority ranking.*

Test Case	Input	Expected Output	Actual Output	Test Case Pass or Fail
User Registration	testuser testpass testname testnumber testemail' 0 test2 0	-New user object is successfully created and stored in the database.	-User object is created and added to database	Pass

User Login:

- *Users will be able to login to an existing account*

Test Case	Input	Expected Output	Actual Output	Test Case Pass or Fail
User Login	abbeycameron password1	-User can successfully login with new account	-user can login with username and password during this instance of the program and new instances	Pass

Create Tasks (BLACKBOX PARTITIONING 1):

- *Users will be able to create tasks and projects.*
- *Users will be able to add tasks to projects.*
- *See test_create_task.py file for the following test cases*

Test Case	**Input**	**Expected Output**	**Actual Output**	**Test Case Pass or Fail**
Create a task	Task Name Deadline Project Name Priority Assigned to (optional) Task Number	-Task has been created/deleted/added to a project -Task/Project deadline has been created	Tasks are successfully created (ones with missing fields fail).	Pass (base case, proper input and successful task addition)
Create a task	<u>One Field Missing Case</u> Deadline Project Name Priority Task Number	-User will get re-prompted to input the missing field.	Task is not created and output fails to be properly captured with missing field	Fail
Create a task	<u>Task Already Exists</u> Task Name Deadline Project Name Priority Task Number	-In this case, the task already exists. -Expect a task already exists output and prompts to enter new information if requested.	Task is not created and user gets to retry (if they want to create a new task).	Pass
Create a task	<u>Task Number Not Updated</u>	-When a task is added to a project, the number in the project should automatically update. -When task is added, number of task in the project should equal the total num of projects in project.	Task number in specified project equals the number of task in that project.	Pass

Create Projects (BLACKBOX PARTITIONING 2):

- *Users will be able to create projects.*
- *Users will be able to add tasks to projects.*
- *See test_create_project.py file for the following test cases*

Test Case	Input	Expected Output	Actual Output	Test Case Pass or Fail
Create a project	Project Name Project ID Deadline Owner Priority Num of Tasks	-Project has been created with all given inputs - since no fields are missing, project should be created no issue	Project is successfully created and added to database	Pass
Create a project	One Field is Missing Project ID Deadline Owner Priority Num of Tasks	-User will not enter in the project name, the expected output would be to re prompt the user to enter the name of the project before continuing.	Test case fails and user is not prompted to make another attempt to fill in missing information.	Fail
Create a project	Project Already Exists Project Name Project ID Deadline Owner Priority Num of Tasks	-Tested using the name of an already existing project. -user will be told project already exists and project cannot be created. -user then prompted to attempt again at inputting information if they request it.	-Ouputs to the user that the project already exists and if they request it, they can re-enter information.	Pass

Assign Task to User:

- *Users can assign tasks to a specific user profile, or have the system automatically assign tasks to users.*
- *See test_assign_task.py file for the following test cases*

Test Case	Input	Expected Output	Actual Output	Test Case Pass or Fail
Assign a task to a specific user	Task User	-User will be assigned to a task -User can see what tasks they are assigned to	User is assigned to task at any time (optional at time of task creation). As of now backend shows all tasks on a project the user's manager owns.	Both pass (one with proper input, and one where user fails to input correct data)

View Profile

- *User should be able to view profiles using data retrieved from the database.*

Test Case	Input	Expected Output	Actual Output	Test Case Pass or Fail
View profile(s)	CLI input: "mollyd", "View", "Exit"	CLI print output: " mollyd Molly Doe 6134444444 1 Collingwood St molly@email.com 2 Olivia Wilson"	CLI print output: " mollyd Molly Doe 6134444444 1 Collingwood St molly@email.com 2 Olivia Wilson"	Pass

Edit Profile

- *User is able to edit their own profile information in the database, but not others.*

Test Case	**Input**	**Expected Output**	**Actual Output**	**Test Case Pass or Fail**
Edit data of own profile	CLI input: 'cameronj', 'Edit', 'name', 'Cameron John', 'Exit'	Name field of user cameronj changed to ‘Cameron John’	Name field of user cameronj changed to ‘Cameron John’	Pass
Edit data of own profile	CLI: input: ‘nonsense’, ‘Exit’	“Unrecognized Input”	“Unrecognized Input”	Fail? User ‘nonsense’ does not exist, thus cannot edit. Intentional fail case.