MSC- COMPUTING

SREEJITA GHOSH  
STUDENT ID – 3008916

Building an Application for Task Management amongst multiple Users (Assignment 2)

**Overview:**

In this assignment, we have created a web application where user can add a Task-board inside which user can invite another user from present in the database. And also, user can add task inside every task-board where any user can assign task to the users those are being invited by the owner of the task-board. The users who are invited in the task-board can edit and delete the task as well as they can remove any user they want except for the owner of the task-board.

**Pages:**

This application consists of both .py and .html and one. yaml page. Details of the page as described below:

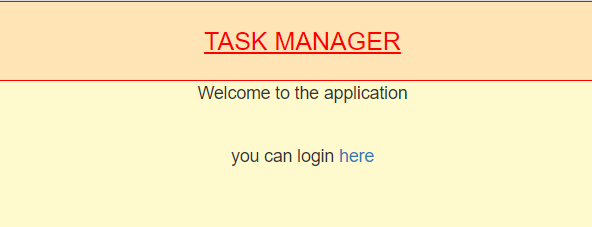
* login-logout.py:

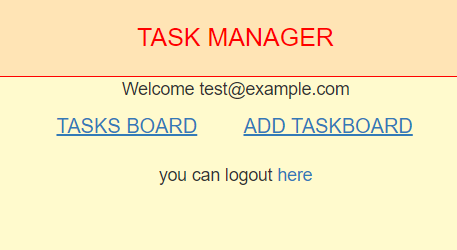
This is the main page of the application as each and every page is being linked with. We have used webapp2 library to receive the response from the web and send response to the web and JINJA library to set up the environment.

Using “MyUser” class I have imported the “user” module, so that we don’t have to create a login-logout page as it is directly linked with the google app-engine. We also have used “ndb.key” functionality to fetch all the data from database.

* login-logout.html:

This is the front page of the application. Here user is allowed to login using their own email id. Once the user is logged in, the user has the option to add a Task-Board and the user can access the task-board where they can see how many task-boards they have created. And in the bottom, there is a logout option.



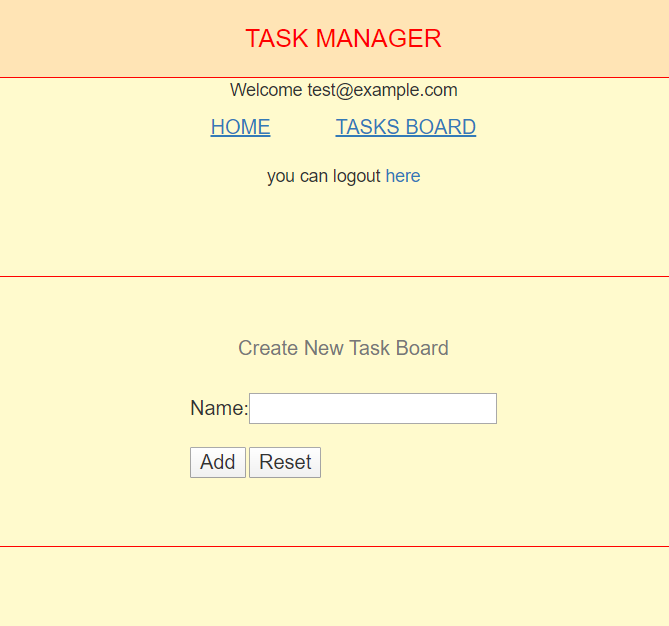


* addTaskBoard.py:

This is the page where the logged-in user can add task-board. While adding the task-board name, database will check whether the same name task-board is already present in or not. If the same name is already present it will not add the task-board and will ask the user to fill it again. And if the task-board is not present then it will directly add the details in the database.

* addTaskBoard.html:

Once the user login, now they can add task-boards in the database. There is only one field where the user needs to provide a task-board name to add a task-board.

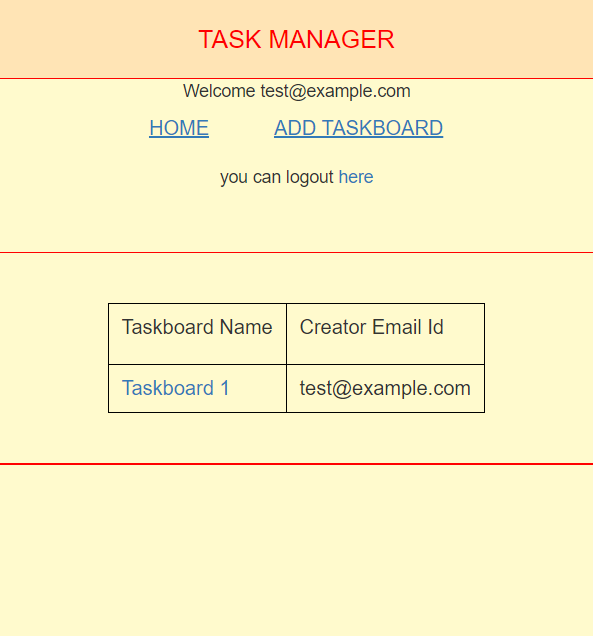


* taskBoard.py:

In this page, the user will be able to see what task-boards are present in the “taskBoarddata” database and displayed. And also, the user can add new task-board to the “taskBoarddata” database. Once the task-board has been created it will show the data in this page and if the user will click the task-board name it will go to the next page.

* taskBoard.html:

In this page, the user will be able to see how many task-boards has been made. And also, the user can add new task-board to the database. Once the task-board has been created it will show the data in this page and if the user will click the task-board name it will go to the next page.



* invite.py:

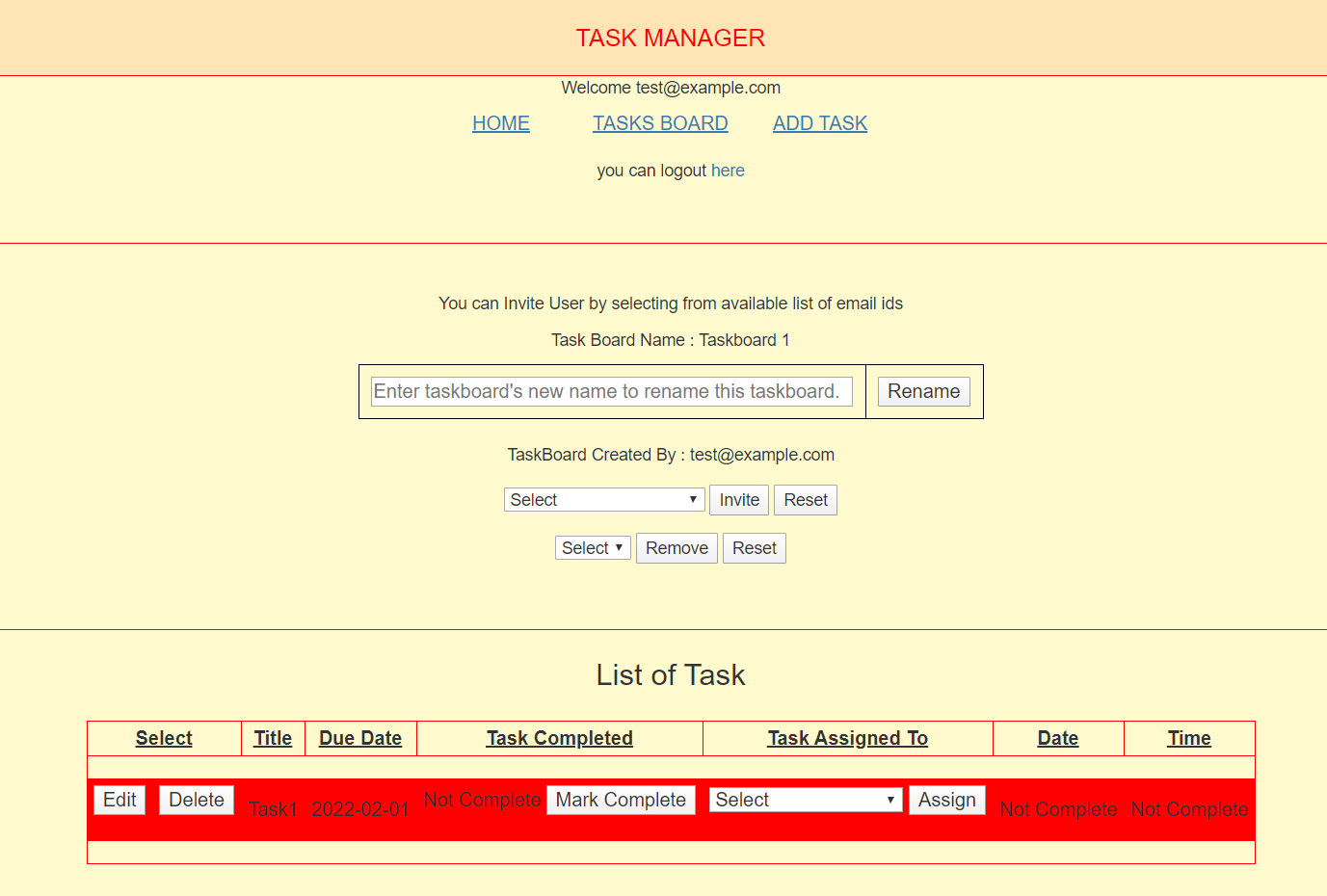
In this page, we are inviting other users present in the “taskBoarddata” database. And adding task in “taskdata” database and then we are fetching the data from the same database to display. We have a feature name Mark Complete, where we are checking whether that task completed in the database. If the task is completed then date and time will display along with it, otherwise the date and time will remain blank.

We have another feature named “Assign”, where the user can assign any user present in the “taskBoarddata” database and once the user is assigned the red highlight will be removed. And if the user is removed from the task and taskboard then the red highlight will remain highlighted.

Then we have the feature to “Edit” & “Delete” for task, where any user can edit and delete the task from the “taskBoarddata” database. And the last feature is to “Remove” task-board from “taskBoarddata” database.

* invite.html:

In this page, the user will be able to add task in the “taskdata” database and the list will of task will be shown. The user can invite another user from “taskBoarddata” “taskBoarddata” database. And the user can assign task to another user, this feature can be done by the invited user. Where everyone can edit and delete task and also remove other users.



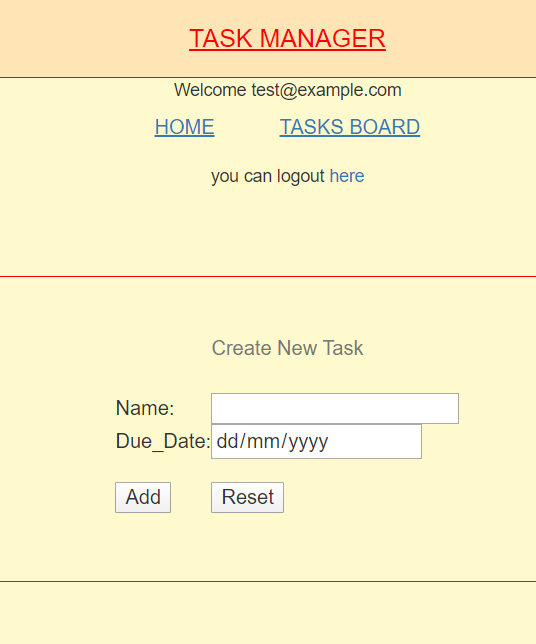
* addtask.py:

This is the page, the owner of the task-board and the users invited can create tasks inside the board. While adding the task name, “taskdata” database will check whether the same name task is already present in or not. If the same name is already present it will not add the task and will ask the user to fill it again. And if the task is not present then it will directly add the details in the “taskdata” database.

And then there is another field named “Due\_Date”, where user have to assign due date for every task they create.

* addtask.html:

Once the user login, now they can add task in the “taskdata” database. There are two field where the user needs to provide a task name and due date to add a task in “taskdata” database.



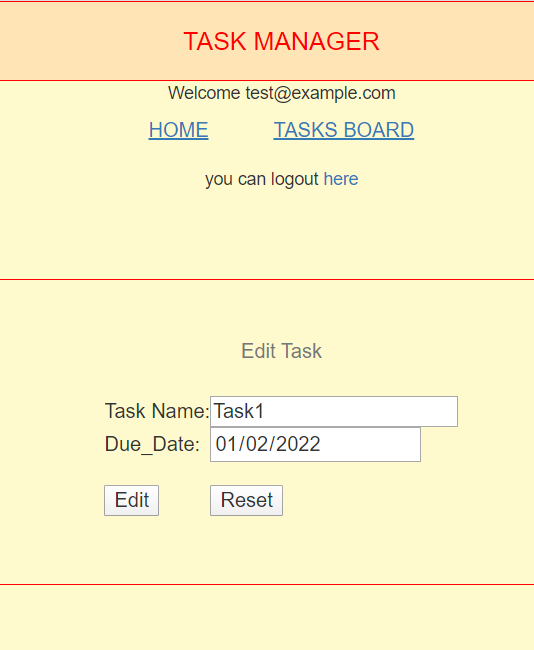
* editTask.py:

This is the page, the owner of the task-board and the users invited can edit tasks inside the board. While editing the task name, “taskdata” database will check whether the same name task is already present in or not. If the same name is already present it will not change the task name and will ask the user to fill it again. And if the task is not present then it will directly add the details in the “taskdata” database.

And then there is another field named “Due\_Date”, where user have to assign due date for every task whenever they edit the task.

* editTask.html:

Once the user login, now they can edit task in the “taskdata” database. There are two field where the user needs to provide a task name and due date to edit a task in “taskdata” database.



* DeleteTask.py:

In this page, the owner of the task-board and the users invited can delete tasks present inside the “taskBoarddata” database.

* AssignValue.py:

In this page, the owner of the task-board and the users invited can assign task to any of the users invited by the owner and their email id displayed in the list of both This is the page, the owner of the task-board and the users invited can edit tasks inside the “taskBoarddata” & “taskdata” database.

* Complete.py: We have a feature name Mark Complete, where we are checking whether that task is present in the database or not. If the task is present then the task will be marked completed and date and time will display along with it.
* Incomplete.py: We have a feature name Mark InComplete it is accessible when a user has already marked a task as completed. User can change it to in complete in that case the task will be incomplete and then the date and time will become blank

**Data-Structure and Model:**

We have used ndb model in this application. ‘ndb’ means non-sql database which includes different classes that has been stored in the database with the required data.

* user.py: In this database, we have stored the data of the users here in the following attribute:
* email\_address: Here we are storing the email of the user which the user has used for login purpose.

* taskboard: We have used this attribute to link the user database with the task-board database. So that whenever a user creates a task-board, the data will store under the user name.
* taskBoarddata.py: In this database, we have stored all the data of the task-board.
* taskBoarddata: Here we are storing the name of the task-boards.
* email\_address: Here we are storing the email address of the users who are being invited and given access to create a task, change the name of the task, delete the task and also assign the users who are being invited by the owner.
* owner: Here we are storing the email address of the users who have created the task-boards.
* taskdata.py: Here we are storing the data of the task including the creating of task.
* Title: Here we are storing the name of the tasks
* Due\_Date: Here we are storing the due date for the task, which we are assigning at the time of creating the task.
* Task\_Completion: Here we are storing whether the task is being completed or not.
* Task\_assigned: Here we are assigning user who are being invited in the board by the owner. And the task can also be assigned by the users already invited in the board.
* Date: Once the user mark that the task is completed, the date will show in the database.
* Time: Once the user mark that the task is completed, the time will show in the database.

**Design Description:**

* Application has a simple UI which will be easily accessible by every user. Each and every page consist of button which will help the user to redirect to the next page and home.
* In the first page user can see buttons for login, logout. And once the user is logged in then have the access for the adding task-board and viewing the task-board.
* In the add task-board user can add task-boards name in the “taskBoarddata” database.
* In the add task user can add task name in the “taskdata” database.
* Both the Owner and invited User can use the edit and delete button.