ASKme.fm

This project is like the website askme.fm. I worked on this project to practice and to learn more :) … enjoy the documentation.

What does it do?!

This is a multi-user CLI application powered by a file-based database. You can ask questions and other people answer you; gust like ASKme.fm - the website-. It has 9 main functions: ask a question, reply, see the other users questions, see your questions, remove a reply, remove a question, quit, logout, and get a specific feed. We will discuss them in more - but charming - details. It also allows creating an account.

Let’s start with the database.

This app. Is powered by a file-based database as we mentioned earlier. The database is located at “files” directory. It is id-based database. Each user, question and reply is identified by an id. We can use and see them - which is not securely fine- for simplicity in this project. There is a file named users.txt stores the users info in this format:

```
User_id user_name user_email password\n
```

Each file of the following: user_id.txt, question_id.txt and reply_id.txt stores the last id that can be used so that each id is unique.

Questions are stored in different way. Each question and its replies are stored in a unique file with this name format:

```
Question_id.txt    for ex; 12.txt
```

In a question file,the question and its replies are stored in the following format:

```
Question_id user_id is_ann title body
  Reply_id user_id question_id body
  .
  .
  .
```

Is_ann is referred to is anonymous question. If is_ann == zero then any user can see and reply to that question, else the only user with the id stored in is_ann can see it.

There is also a file named questions.txt , this file stores metadata about each question like it's id, user_id and is_ann for faster search.

Let's look at the source code

The frist used function in the code is append_to which take the path to a file in the database and vector of queries to append them to the file.

Then get_id which return a fresh id to be used. It only takes the path to the id file.

Belong_to which takes a query and check if it belongs to a specific scheme.

Remove_from which remove a query from the database.
Put_to do as the append function but to a single query.

Then the contaner class. This is the main core of the app. Its constructor function contain a loop which run a specific member function due to the user input. Compile the source code and run it for more understanding. We have the user variable which contain some useful metadata about the user who is on now.
Lets look at the member functions:
Login function looks for the user email and password in the database and makes sure that the user has an account.

Logout it assigns -1 to the user_id so he/she is logged out.
Sign up: it takes the user data and makes a new account.

Show all feed: which show the questions that are anonymous - every one can see it - and the questions that are directed to you -the user-. It scanes all the questions in the database to show the appropriate ones.

Add question: it takes the anonymous variable data (id if the question is directed to some one specifically 0 for every one) and the question statement in one line as following:

```
Question subject; question body
```

And add the question to the database.

Reply to: takes the id of the question that the user want to reply to and the reply in one line as following:

```
Question_id reply statement
```

Then checks if there is a question with this id and if the user is allowed to see and reply to this question.

Remove reply: takes question_id and reply_id, checks if this reply belongs to this user and removes it.

Remove thrind: takes the question id, checks if it belongs to this user and removes it with its replies.