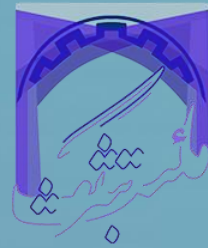


مکتب شریف

اولین بوتکمپ آموزشی - استخدامی ایران



PYTHON

BOOTCAMP

PRACTICE 9



Building a Contact Management System with User Authentication

In this exercise, you will build a contact management system that requires users to authenticate before they can add, edit, or delete contacts. You will use Python's built-in user authentication system to manage user accounts, and you will use OOP concepts to organize your code. You will also use the Pickle module to save the contacts to a file, and you will package the application for distribution using setuptools. Finally, you will use regular expressions (regex) to validate user input.

Step 1: Design the Contact and User Classes

Create a Contact class that has the following attributes:

- name (string)
- email (string)
- phone (string)

The class should also have methods for adding, editing, and deleting contacts.

Create a User class that has the following attributes:

- username (string)
- password (string)

The class should also have methods for creating, authenticating, and modifying user accounts.

Step 2: Implement the Contact Management System

Write a script that allows users to interact with the Contact class after they have authenticated. The script should have the following options:

- Add a new contact
- Edit an existing contact
- Delete a contact
- View all contacts
- Quit

When the user chooses to add or edit a contact, the script should prompt the user for the name, email, and phone number. Use regex to validate the email and phone number inputs.

When the user chooses to delete a contact, the script should prompt the user for the name of the contact to delete.

When the user chooses to view all contacts, the script should display a list of all contacts.

Before allowing the user to access the contact management system, the script should prompt the user to authenticate by entering their username and password.

Step 3: Save Contacts and Users to a Separate Files

Use the Pickle module to save the contacts and users to separate files. When the script starts up, it should load the contacts and users from the files if they exist.

Step 4: Implement User Authentication

Use Python's built-in user authentication system to manage user accounts. When the script starts up, it should prompt the user to create a new account or authenticate with an existing account. Once the user has authenticated, they should be allowed to access the contact management system.

Step 5: Package the Application

Choose appropriate packaging for your application. You can use this packaging tree.

```
contact_manager/  
├── __init__.py  
├── main.py  
├── contacts.py  
├── users.py  
├── validators.py  
└── data/  
    ├── contacts.pickle  
    └── users.pickle
```

Step 6: Add Additional Functionality

To make the contact management system more useful, consider adding some additional functionality. For example, you might add the ability to search for contacts by name or email address, or you might allow users to group contacts into categories.

You could also add support for exporting contacts to a CSV file or importing contacts from a CSV file. Another option would be to allow users to set reminders or notes for each contact.

Step 7: Test the Application

Verify that the application works as expected and that users are prompted to authenticate before accessing the contact management system.

نکات

- مهلت ارسال تمرین تا **پایان ساعت ۲۴ روز چهارشنبه** می باشد.
- زین پس تمامی تحویل تمارین تنها و تنها از طریق **گیتهاب (Github)** صورت می پذیرد.
- بدین منظور لازم است یک مخزن (**repository**) **خصوصی (private)** ساخته شود.
- آدرس ریپازیتوری را در کارتابل شخصی خود اعلام کرده و تیم تدریس را بعنوان **collaborator** بیفزایید:
 - [فاطمه علی میرزائی](#)
 - [حنانه فلاح](#)
 - [مهدی میرزائی](#)
 - [میلاذ حسین زاده](#)
- ملاک و معیار ارزیابی **تاریخ آخرین commit** شما می باشد . (بصورت استاندارد و اصولی کامیت انجام شود).
- **توصیه دوستانه.** از مواجهه با هیچ سوالی نترسید. به هر میزانی که در حل سوالات پیشروی کرده باشید نمره بخش مورد نظر را دریافت می کنید. بنابراین بیش از آنکه رسیدن به خروجی نهایی مهم باشد، تلاش شما ارزشمندتر است.
- قطعا هدف از تمارین صرفا رسیدن به جواب نهایی نیست و تمیز بودن کد و خلاقیتی که در انجام آن به خرج می دهید از اهمیت و امتیاز بالایی برخوردار است. ارائه راه حل کلی و عمومی برای یک مسئله که حالت های مختلف آن را در نظر بگیرد و فراتر از خواسته ی مسئله است. (خواسته ی مسئله گسترش داده شود یا حالت های خاص مسئله را پوشش دهد. قطعا مشمول امتیاز بیشتری خواهد شد).
- سوالات امتیازی شامل مواردی است که نیازمند سرچ بیشتر شما عزیزان می باشد. بنابراین حل این سوالات نمره امتیازی دارد.

موفق باشید