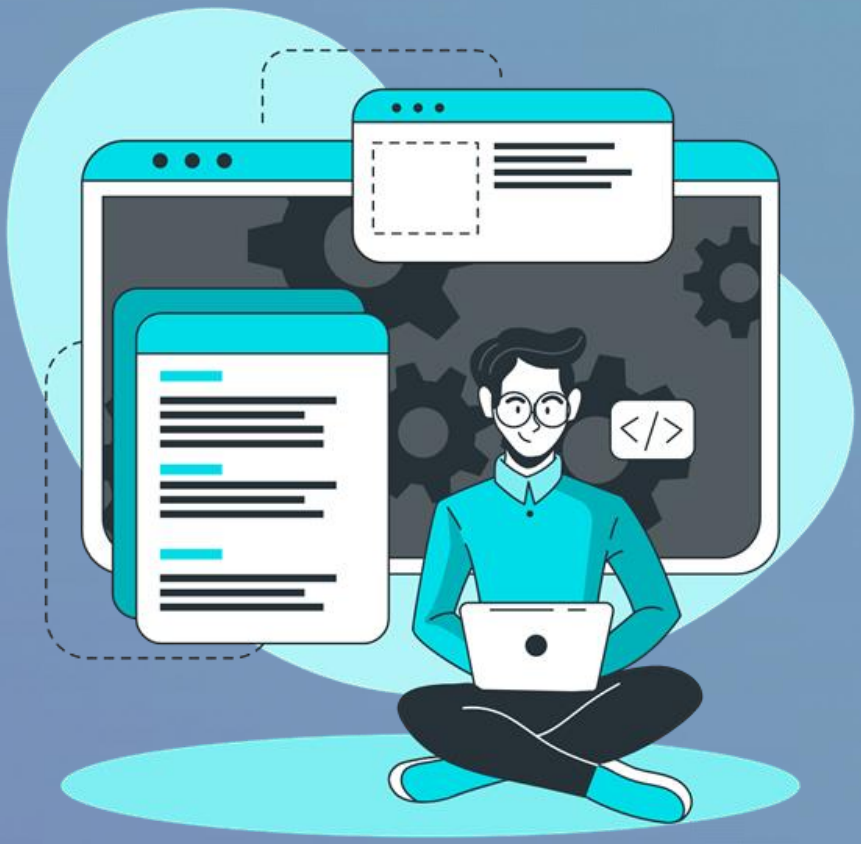
مکتب شریف

اولین بوتکمپ آموزشی - استخدامی ایران

PYTHON

BOOTCAMP

Project1-1



Introduction

In this document, we'll outline the design and implementation of a bank system using Object-Oriented Programming (OOP) principles in Python. The system will allow users to register, login, create accounts, deposit, and withdraw and transfer funds. PostgreSQL will be used as the database in backend, managed by a DBManager class utilizing Python's context managers. Additionally, logging will be employed to record transaction processes (use Decorators).

To facilitate collaborative development and version control, developers must use Git and GitHub effectively. The project repository should be organized following a Git branching model, such as Gitflow.

Phase 1

Git Configuration

First, create a Git repository on GitHub.com. Invite your mentors to collaborate. Create a main branch for the final version of the project and a develop branch to commit all changes.

Branches:

- main: Represents the stable production-ready codebase.
- develop: Serves as the main development branch, containing the latest changes.

System Overview

The bank system consists of the following components:

- User Management: Registering and logging in users.
- Account Management: Creating accounts, depositing, and withdrawing funds.
- Database Management: Storing user and account data in PostgreSQL.
- Logging: Recording transaction processes for auditing and debugging.

Object-Oriented Design

The system will be designed with the following classes:

1. User: Represents a bank user with properties such as username, password, and account details.

2. BankAccount: Represents a bank account associated with a user, handling deposit and withdrawal and transfer operations.
3. DBManager: Manages database operations using Python's context managers for connection management and transaction handling.

Database Schema

The PostgreSQL database will consist of the following tables:

- users: Stores user registration details.

  Columns: user_id, username, password_hash.

- accounts: Stores bank account details.

  Columns: account_id, user_id, balance.

- trasnaction: store transaction between accounts.

  Columns: transaction ID, account ID, transaction type (deposit, withdrawal, transfer), amount, timestamp, etc.

Logging Configuration

The logging module will be configured to record transaction processes, including account operations such as deposit and withdrawal. Logs will be stored in a dedicated log file with appropriate formatting and log levels for easy interpretation and analysis.

** **Notes:** Exception handling is crucial for robust error management in any project. Developers should follow best practices to raise and handle exceptions effectively throughout the project

نکات

- مهلت ارسال تمرین تا **پایان ساعت 9 روز پنج شنبه** می‌باشد.
- زین پس تمامی تحویل تمارین تنها و تنها از طریق گیتهاب (**Github**)صورت می‌پذیرد.
- بدین منظور لازم است یک مخزن(**repository**) بصورت (**private**) ساخته شود.
- آدرس ریپازیتوری را در کارتابل گروهی خود اعلام کرده و تیم تدریس را بعنوان **collaborator** بیفزایید:
  - **غزاله رنجبران**
  - رضا غلامی
  - زهرا متین فر
  - سعید کرمانشاهی
  - سینا مبارز
  - سید محمد رضوی

- ملاک و معیار ارزیابی تاریخ آخرین **commit** شما می‌باشد . (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 می‌کنید. 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 خواسته‌ی مسئله است. (خواسته‌ی 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 می‌باشد. بنابراین حل این سوالات نمره امتیازی دارد.

موفق باشید