Summer Python Course Final Project

A. Python Code duplication checking tool

1. Introduction

Design and implement a simple Python code duplication check tool that can check batches of code for duplication and display the results.

2. Basic Requirements

a. You need to use GUI libraries such as Tkinter and PyQt5 to design GUI, or use front-end and back-end frameworks supported by Python for development and design.

b. A Python code duplication checker that can batch import code, check for duplicate code, and display duplication rate.

c. The interface must be beautiful, and other functions can be added by yourself, but it must be simple and easy to use.

3. Required Tasks

a. Login: Record each login time and display it in the GUI.

b. Query: The ability to perform bulk import of Python code functions and one-to-many queries in the same directory.

b. Code duplication check function: giving the repetition rate between two codes, and displaying the duplication check results in a reasonable and beautiful manner in the GUI in descending order of repetition rate.The duplicate checking function used in the project needs to be researched by yourself.

c. Clicking on each result entry can display the contents of the two codes in detail, and highlight the similar and different parts in the displayed code.

4. Optional Tasks

a. History function: Store each imported duplicate check result, and access the past duplicate check results. No need to perform duplicate check calculations and directly display the duplicate check results.

b. Management function: Manually identify plagiarism. For the duplicate check results of every two codes, add the manual labeling function of "plagiarism judgment" and the function of exporting the list of suspected plagiarists， and it can be linked with "b. History function ", and the suspected plagiarism information is also stored as information in the history record.

c. Export: All plagiarized codes can be exported.

d. Clustering: You can choose to conduct a one-to-many or group self-check query. And when the group self-checks, the query results are divided into several groups according to the plagiarism situation and displayed visually.

B. The Taste of Beihang - What to Eat at Beihang University

1. Introduction

Beihang University has numerous cafeteria food counters, but students often only stick to a few counters and become accustomed to certain dishes. Meanwhile, many students also face difficulties with choosing what to eat in the cafeterias. Design a tool that allows the input of food dishes and provides recommendations to users based on taste preferences, keywords, and previous meal history.

2. Basic Requirements

a. Use GUI libraries such as Tkinter, PyQt5, or Python-supported frontend and backend frameworks for GUI design and development.

b. Beihang University dishes must be actual food dishes found in Beihang cafeterias, and dish recommendations should be made using relevant recommendation algorithm.

c. The interface should be aesthetically pleasing. Other features can be added as needed, but simplicity and user-friendliness should be ensured first and foremost.

3. Required Tasks

a. CRUD: Implement CRUD (Create, Read, Update, Delete) operations for Beihang University's cafeteria halls, food counter, and dishes. At least 150 real dishes should be catalogued, including breakfast, beverages, and main meals.

b. Record: Allow users to record the dishes that they have eaten (supports addition, deletion, and modification).

c. Favorites: Allow users to bookmark specific dishes, food counters, and cafeteria halls.

d. Recommendation: Provide meal recommendations based on time of day, user taste preferences, and past meal history for breakfast, lunch, and dinner.

e. Comments: Allows users to place comments on dishes and support interactive communication among multiple users.

f. Management: Implement a user support center that supports registration, login, and personal information management.

4. Optional Tasks

a. Earn 1 point for every additional set of 150 real dishes added (up to maximum of 3 points).

b. With the help of large language models (LLMs), provide word suggestions where if a user inputs a word, the user would receive dish recommendations related to that word. For example, if a user inputs the keyword "warm," the program would recommend dishes like braised pork, based on some recommendation algorithms.

c. Add other functionalities based on own practicality and workload, and give scores to the functions accordingly.

C. Simplified stock market data visualization and analysis evaluation

1. Introduction

The stock market is a complex dynamic system involving a large amount of data and information. For investors, how to effectively obtain, display and analyze stock market data is an important and difficult problem. This topic aims to design and implement a stock market data visualization and analysis evaluation tool. It can use Python libraries to display and analyze data based on the data set of 100 stocks that have changed in 1 year. If you have the ability, you can integrate various types of data and conduct a comprehensive analysis and evaluation of stocks.

For the relevant data used in the question, please contact the teaching assistant via WeChat to obtain it.

2. Basic Requirements

a. You need to use GUI libraries such as Tkinter and PyQt5 to design GUIs, or use front-end and back-end frameworks supported by Python for development and design.

b. Realize the visualization of stock market data analysis.

c. The interface design should be simple and beautiful, and the operation should be intuitive and easy to use.

3. Required Assignment

a. Design a simple and beautiful GUI. The requirements are as follows:
   - Users can preview the drawing results on the GUI.
   - You can select Excel file import on the interface
   - You can search for stock codes, select stocks to be analyzed and display visualization images of corresponding stocks.
   - Allow users to preview different images according to their needs, display and visualize results.

b. Read in Excel data tables; perform data statistics, and draw relevant data of stocks, perform data statistics in 8 data points including opening price, closing price, highest price, lowest price, trading volume, increase or decrease, amplitude, and turnover rate, calculate the average value, sum or maximum value, and minimum value of the data of each stock in the year, and use line charts, bar charts, pie charts, scatter charts, etc. to visualize the calculation results. Please select at least 5 pictures for drawing to reflect the overall situation of the stock market that year. Available pictures:
   - Opening and closing price average bar chart.
   - Total trading volume bar chart.
   - Bar chart of the highest price reached by each stock.
   - Bar chart of the lowest price reached by each stock.
   - Bar chart of the compound increase or decrease of each stock in the year.
   - A scatter plot of the average amplitude of each stock.
   - A bar chart of the average turnover rate of each stock.

c. For a single stock, you need to use the stock's opening and closing prices to draw a price line chart of the single stock. In addition, draw the 5-day (short-term), 10-day (medium-term), and 30-day (long-term) moving averages and place them in the same picture, using different colors or line types to distinguish them, and mark them at the golden cross and death cross points.

d. Read the trading volume, price increase or decrease, amplitude, turnover rate and other data of a single stock, and select appropriate graphics for visualization, such as a bar chart of trading volume, a scatter plot of price increase or decrease, a line chart of amplitude, a line chart of turnover rate, etc.

e. Use the opening, closing, highest and lowest prices in Excel to draw a candle chart for each stock, and draw the 5-, 10-, and 30-day moving averages in the picture at the same time.

4. Optional Assignment

a. 1. Use a combination of K-line charts and other images to analyze stocks. You need to integrate the above images into a complete K-line chart and embed it in the GUI together with other related stock data analysis charts.

b. Login and registration functions to distinguish users.

c. Record each login time and display it in the GUI.

d. Add history record function: store each visualization result, and access past results. No need to perform stock analysis again, but directly display the visualization result of the operation.

D. Task Schedule

1. Introduction

College life is rich and colorful, with homework and activities intertwined, so you need to design a task scheduling app to record tasks reasonably and help students plan their time better.

2. Basic Requirements

a. You need to use Tkinter, PyQt5 and other GUI libraries to design the GUI.

Note: Since this task is relatively simple, you must **use a Database** and **describe the database usage of this task in detail in the homework report**.

b. Complete the task writing system, calendar, task status distinction, historical data review, task scheduling and other functions.

c. The interface is required to be beautiful. Other functions can be added by yourself, but it must be simple and easy to use.

3. Required Assignment

a. User login and personal information system: fill in basic user information, you can design the items that need to be filled in by yourself, and use the database to store.

b. CRUD: (Create, Read, Update, Delete) operations of tasks, including titles, content descriptions, deadlines, importance, etc., and you can also allow tasks to be modified and deleted.

Note: Database storage is required.

c. Task status distinction. The task status of unstarted, ongoing, completed, and expired tasks is clearly distinguished in a visual form.

d. Task completion confirmation. When the task is completed, the user can use a certain method, such as ticking to eliminate the task.

e. Task reminder: Filter and display tasks according to the time from the task, especially to display tasks that need to be completed today, and to clearly show whether the task is urgent in a visual form.

f. Calendar system. You can view the task schedule for each date of a certain month by browsing the calendar.

g. Task scheduling. By setting a task list and providing idle time, the system will automatically arrange the task plan in a certain task scheduling method. The task scheduling method can be designed by yourself as long as it is reasonable.

4. Optional Assignment

a. Task scheduling with weights. By setting a task list, setting priority weights for tasks, and giving free time, the task plan is automatically arranged.

b. Support daily tasks. For daily tasks, you only need to set them once, and they will automatically appear in the task list every day within the specified time.

c. Task category division, set task categories, such as sports, learning, etc.

d. Task review. Able to perform visual data analysis based on historical task completion data.

E、Shared Exercise Platform

1. Introduction

Please design a platform where students can upload and share questions. Students should be able to test themselves on it.

2. Basic Requirements

a. Use GUI libraries such as Tkinter, PyQt5, or other front-end and back-end frameworks Python supported.

b. The questions should include various formats such as multiple choice and fill in the blanks. Obtain questions on your own.

c. The interface should be aesthetically pleasing but not overly fancy to distract from problem-solving. You can add additional features as desired, but they should be user-friendly and easy to use.

3. Required Assignment

a. Basic Requirements: User and administrator registration, login, and personal information management.

b. User Groups: Users can choose to create and join groups, and users can search and join groups voluntarily.

c. Upload: Recognize the text in PDF or pictures automatically. After recognition, the extracted text results can be edited to complete the input of questions. (Hint: Use OCR)

d. Question Groups: Design your own or utilize existing data structures to organize questions into categories based on chapters or other criteria. When solving problems, users can choose a specific group of questions to work on. The problem-solving interface should be designed according to individual preferences without excessive requirements.

e. Question Sharing: Users can choose to share a group of questions with a specific group or make them available to everyone. The recipients of the shared questions gain access to the question group.

f. Search for Groups: The search should have customizable parameters, but the search scope should include shared question groups and the user's uploaded questions. It should not search for question groups that have not been shared.

g. Error log: Based on the user's incorrect answers, frequency of errors, and the user's specified subject and question quantity, consult relevant recommendation algorithms to generate a set of questions that the user should prioritize re-solving using a scientifically effective algorithm of your choice.

4. Optional Assignment

a. The system have the responsibility of screening sensitive words and removing them from the question bank. Find a way to implement this functionality.

b. Visualize student abilities. Based on the type and time of incorrect answers, refer to relevant materials and define a conversion standard from student's incorrect question information to student's ability information. Create a graph showing the change in student abilities over time.

c. Implement additional features as desired, earning extra points based on practicality and workload.

F、Autonomy

1. Introduction

a. Design a GUI using GUI libraries such as Tkinter, PyQt5, or other front-end and back-end frameworks Python related such as Django, Flask.

b. Before the system design, write a proposal by yourself, which should refer to the description above.

c. The proposal is expected to be sent to the mailbox sy2206143@buaa.edu.cn in PDF form before 23:55 on the 13th. We will give feedback at 23:55 on the 14th. Do it after the proposal pass.