

CS307 Project2

Members

Team of 12010903林雨航, 12012338曾宪清 in Lab 2.

	林雨航	曾宪清
Basic	function 2~5 funtion 13 one step export	function 1 function 6~12 one step input
enhance APIs	8 new APIs	
better query		basic functions accept no parameter
front-end		using flask in python with http protocol
GUI		show all functions on website page
pool		apply connection pool
privilege	different for top, salesman, manager	
score	50%	50%
python	3.10	3.6.3

hardware

林雨航

设备名称	LAPTOP-EOE1H7A
处理器	AMD Ryzen 7 4800H with Radeon Graphics 2.90 GHz
机带 RAM	16.0 GB (15.4 GB 可用)
设备 ID	
产品 ID	
系统类型	64 位操作系统, 基于 x64 的处理器
笔和触控	没有可用于此显示器的笔或触控输入

曾宪清

设备名称	DESKTOP-KOBRUV5
处理器	Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz 2.59 GHz
机带 RAM	8.00 GB (7.80 GB 可用)
设备 ID	
产品 ID	
系统类型	64 位操作系统, 基于 x64 的处理器
笔和触控	没有可用于此显示器的笔或触控输入

Python packet used:

packet name	version
Flask-Bootstrap	3.3.7.1
Flask-WTF	1.0.1
Flask	2.1.2
psycopg2	2.9.3
WTForms	3.0.1

Basic API

importFour()
Import four information files in one function

stockIn()
Support action of increasing the product inventory according to .csv file

placeOrder()
Place an order which sells specific products to an enterprise according to .csv file

updateOrder()
Update orders in a series of limitations according to .csv file.

deleteOrder()
Delete orders following a series of limitations according to .csv file. No contract delete if order do not exist.

getAllStaffCount()
Return the numbers of people for all types of staffs.

getContractCount()
Return the total number of existing contract.

getOrderCount()

Return the total number of existing orders.

getNeverSoldProductCount()

Return the number of models that are in stock but have never been ordered.

getFavoriteProductModel()

Find the models with the highest sold quantity, and the number of sales.

getAvgStockByCenter()

Calculate the average quantity of the remaining product models for each supply center.

getProductByNumber(model_number: str)

- model_number: a number of specific model

According to input model number, Return the current inventory capacity of each product model in each supply center.

getContractInfo(contract_number: str)

- contract_number: a specific number of a contract.

Return specific contract information in requirements of a specified contract.

oneStepExport(product_number_list: list[str], contract_number_list: list[str])

- product_number_list: list of input parameters in `getProductByNumber` .
- contract_number_list: list of input parameters in `getContractInfo` .

Export required information into local file.

Advanced APIs

getBestSalesman(type: str)

- type: can be one of three strings: 'amount', 'model', and 'order'.
- if type == 'amount':

Return the id and name of salesman selling out most money.

if type == 'model':

Return the id and name of salesman selling out most model number.

else: (type == 'order' or something else)

Return the id and name of salesman selling out most order number.

- Set one more prize in year-end bonus.

getBestProfitProductModel()

- Consider if all models are sold out, return the model name, profit, and remain storage product number. If there are paralleling models, show them all.
- Offer advice on selling strategy: try to sell all output model out.

getOrderBetweenDates(date1: str, date2: str, enterprise: str, contract_num: str)

- date1: in format "year-month-day"
- date2: in format "year-month-day"
- enterprise: the name of client enterprise. Can be empty, regard wrong input as empty.
- contract_num: the number of specific contract. Can be empty, regard wrong input as empty.
- Return all orders (in this client enterprise) (in this contract) between two input days.
- To prevent client blackmail company, and offer a query function for employees.

getMostModelEnterpriseRecent()

- Return the first ten enterprise name making deals with company most frequent.
- Every year to find an excuse to send small gifts to enhance the relationship, stimulate customers to spend more money.

getLossProductModel()

- Return all negative profit model in all already sold-out models, and the model having not been ordered in one year after stock in.
- These model do not have sales market, consider stock in less or none of them in future.

getOrderEachMonth(center: str)

- center: Can be empty, regard wrong input as empty.
- Return the number of orders in every months (in a specific supply center). An order is consider to be done if and only if its contract type is "Finished".
- Top handlers in company can adjust human resource to each supply center, move some employees from less-ordered-center to more-ordered-center.

getProfitBetweenDates(date1: str, date2: str)

- date1: in format "year-month-day"
- date2: in format "year-month-day"
- Return cure profit between two days of whole company.
- To query bills and profit.

setNowDate(date: str)

- date: in format "year-month-day"
- Set present date into specific date. All contract finished after this day will be set "contract_type" into "Unfinished". All contract finished on or before this day will be set "contract type" into "Finished".
- To simulate natural time flow in reality.

Other advanced points

User privilege

Three privileges are designed.

- Salesman: Start with character 's' followed by salesman's number. Like 's12010001'. Can only query orders belonging to him/her.
- Contract manager: Start with character 'c' followed by manager's number. Like 'c12010001'. Can only query orders belonging to his/her contract, and contracts belonging to him/her.
- Top user: Start with character 't' followed by a serial number. Like 't123123'. Designed for senior executive in company, have all privileges.

Respective APIs are as follows:

addSalesmanUser(salesman_num: str)

- salesman_num: a staff number of one salesman.
- Add a user for this salesman, the name of user is created by adding character 's' before this number. If this salesman is already has privilege, do nothing.

addManagerUser(manager_num: str)

- manager_num: a staff number of one contract manager.
- Add a user for this manager, the name of user is created by adding character 'c' before this number. If this manager is already has privilege, do nothing.

addTopUser(user_number: str)

- `user_number`: a serial number of one executive.
- Add a user for this executive, the name of user is created by adding character 't' before this number. If this executive is already has privilege, do nothing.

changeUser(user_name)

- `user_name`: a user's name
- Change privilege into input user. If this user name do not exist, then do nothing. This function is used to simulate human resource management system in company.

Back-end server

Flask is a web framework in Python. Flask server is defined as server software that is capable of running HTTP requests on the public world wide web, private LAN, and private WANs and dedicatedly working for running the software application on the worldwide web.

In project2, we use Flask to build a web application.

Front and back-end interaction

- We use `route()` decorator in Flask to bind URLs to functions.
For example, the URL `'/'` rule is bound to the `home()` function and. Then, if the user visits <http://localhost:5000/home>, the output of the `home()` function will be rendered in the browser.
- By default, Flask routes respond to GET requests. If we want to change the default behavior of the URL `'/'` rule to respond to POST requests, we can do so by adding the `methods = ['POST']` argument to the `route()` decorator.
- We use `request.form` to get input from user and call the database API implemented by ourselves with the inputs as arguments.
- We use `render_template()` function in Flask to render written HTML templates. Then the return value of the database API function is rendered in the browser.

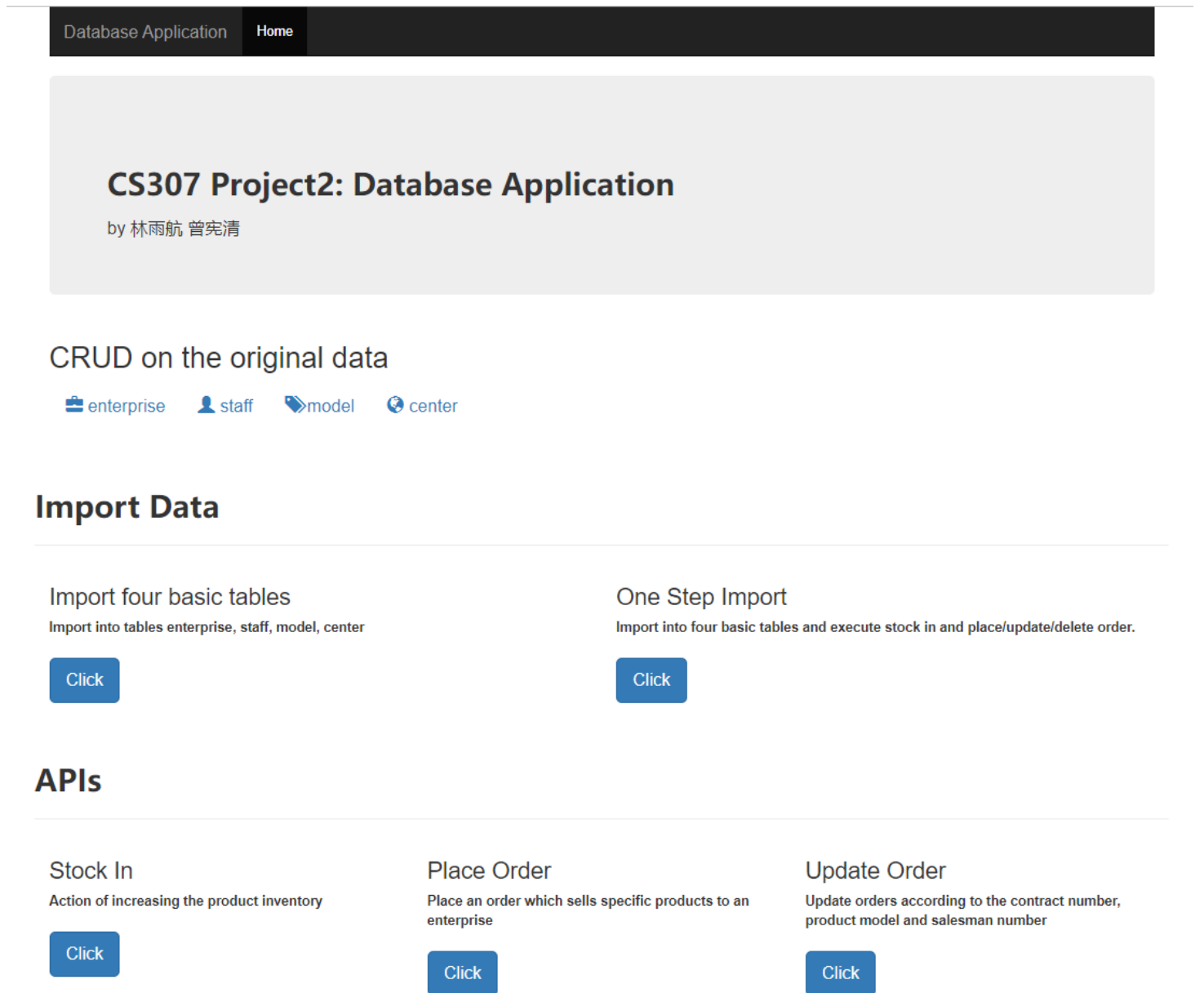
Connection pool

We import the module `psycopg2.pool` and use `psycopg2.pool.ThreadedConnectionPool()` to create a connection pool.

- Once we need to create a new connection, we use `conn = pool.getconn()` to get a connection from the pool.
- When we are done with the connection, we use `pool.putconn(conn)` to return the connection to the pool.
- If the program ends, we use `pool.close()` to close the connection pool.

GUI

Based on the Flask, we decorate the page with Bootstrap3. For example, we use `@app.route('/')` to bind the page we need and return the existed static templates with arguments. The homepage is as followed, for other Web reveal results, we save them in the attached .png file.



Delete Order

Delete orders according to the contract number, salesman and sequence

Click

Get Order Count

Return the total number of existing orders.

[View details »](#)

Get AvgStock By Center

For each supply center, calculate the average quantity of the remaining product models.

[View details »](#)

getBestSalesman

Return the best salesman with the largest sales/product/order amount.

Type

amount

Search

Get All Staff Count

Return the numbers of people for all types of staffs.

[View details »](#)

Get Never Sold Product Count

The number of product_models that are in stock but have never been ordered (sold).

[View details »](#)

Get Product By Number

Find a product according to the product number.

Product Number

Search

Get Best Profit Product Model

Return the product model with the best profit.

[View details »](#)

Get Contract Count

Return the total number of existing contracts.

[View details »](#)

Get Favorite Product Model

Find the models with the highest sold quantity, and the number of sales.

[View details »](#)

Get ContractInfo

Find a contract with a contract number and return the content of the contract.

Contract number

Search

Get Order Between Dates

Return the orders between specific dates.

date1

date2

enterprise

Contract number

Search