**ფინალური გამოცდა (40 ქულა)**

AF-SM-04\_05.2020.G

|  |  |
| --- | --- |
| **სასწავლო კურსის სახელწოდება:** |  |
| **ლექტორი:** |  |
| **სტუდენტის სახელი და გვარი** |  |

**შუალედური გამოცდის დეტალური ინსტრუქცია**

*მოცემულია 3 სავარჯიშო. პირველ და მეორე დავალებაზე დაწერეთ შესაბამისი კოდი პითონის ფაილში. ორივე დავალება შეასრულეთ* ***ცალ-ცალკე ფაილში.*** *მესამე დავალება შედგება 5 ღია შეკითხვისგან. დავალებების შესრულების შემდეგ ატვირთეთ ორივე დავალების* ***py ფაილები*** *და* ***აღნიშნული word ფაილი, სადაც ღია შეკითხვებზე თქვენი პასუხებია მითითებული.*** *გადაამოწმეთ ატვირთული ფაილების სისწორე. გისურვებთ წარმატებებს!*

**სავარჯიშო 1 (12 ქულა) - Parser:**

მოცემულია **sample.html** ფაილი (იხილეთ თანდართული ფაილი). შეასრულეთ შემდეგი დავალებები:

* **(1 ქულა)** დაწერეთ კოდი, რომელიც წაიკითხვას html ფაილს.
* **(3 ქულა)** Beautiful Soup მოდულის გამოყენებით html კოდში მოძებნეთ სასწავლო პროგრამები და ჩამონათვალი შეინახეთ ცალკე სიის სახით.
* (**3 ქულა**) ცალკე ცვლადში შეინახეთ html-ში მოთავსებული BTU-ს ლოგოს ლინკი
* **(5 ქულა)** აღნიშნული ლინკის და requests მოდულის გამოყენებით, დაწერეთ კოდი, რომელიც წამოიღებს სურათს ლინკიდან და შენახავს მას png ფორმატში თქვენს საქაღალდეში.

import os

from bs4 import BeautifulSoup

os.system('py   -m pip install bs4')

with open('sample.html') as file:

    soup = BeautifulSoup(file,'html.parser')

list = []

div = soup.find('div',{'id':'programs'})

li = soup.find\_all('a')

for i in li:

    list.append(i.text)

# print(i)

footer = soup.find('footer',{'id':'footer'})

logo = footer.find\_all('p')

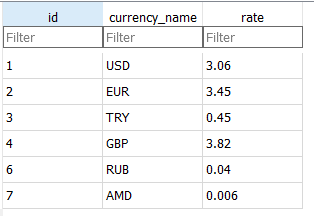
p = logo[1].img['src']

print(p)

**სავარჯიშო 2 (13 ქულა):**

მარტივი ვებ აპლიკაცია მოცემულია **app.py** ფაილში (იხილეთ თანდართული ფაილი). აღნიშნულ ფაილში მოახდინეთ ბაზის ინტეგრირება და დაწერეთ კოდის ფრაგმენტი შემდეგი პირობების გათვალისწინებით.

* **(3 ქულა)** აღწერეთ currency ცხრილის მოდელი კლასის სახით Flask-SQLAlchemy მოდულის გამოყენებით (Currency ცხრილში მონაცემები წარმოდგენილია შემდეგი სახით - იხ. ქვედა სურათი)
* **(3 ქულა)** კლასში დაამატეთ \_\_str\_\_() მეთოდი, რომელიც აღწერს ობიექტს სტრიქონის სახით.
* **(3 ქულა)** დაწერეთ ბრძანება, რომლის მეშვეობითაც შექმნით მონაცემთა ბაზას და ცხრილს.
* **(4 ქულა)** დაწერეთ ბრძანება, რომლის მეშვეობით დაამატებთ ახალ ჩანაწერს ბაზაში.

****

from flask import Flask

from flask\_sqlalchemy import SQLAlchemy

app = Flask(name)

app.config['SECRET\_KEY'] = 'asdsa'

app.config['SQLALCHEMY\_DATABASE\_URI'] = 'sqlite:///currency.sqlite'

db = SQLAlchemy(app)

class Currency(db.Model):

    id = db.Column(db.Integer,primary\_key = True,autoincrement=True)

    currency\_name = db.Column(db.String(20),nullable=False)

    rate = db.Column(db.Float,nullable=False)

    def \_\_str\_\_(self):

        return f"valutaa{ self.currency\_name} , romlis kursia {self.rate}"

with app.app\_context():

    db.create\_all()

    usd = Currency(currency\_name='USD',rate=2.4)

    eur = Currency(currency\_name='eur',rate=22)

    db.session.add(usd)

    db.session.add(eur)

    db.session.commit()

@app.route('/')

def hello\_world():

    return 'Hello World!'

if name == 'main':

    app.run()

**ღია კითხვები (15 ქულა):**

1. რა განსხვავებაა GET და POST request-ებს შორის. მოიყვანეთ შესაბამისი მაგალითები.

GET გამოიყენება მაშინ, როდესაც request-ის გაგზავნა ხდება URL-ის მეშვეობით. შესაბამისად, URL-ში ეთითება

საჭირო პარამეტრები (პარამეტრის სახელი და მნიშნველობა). გამოიყენება ‘&’ სიმბოლო პარამეტრების

ერთმანეთისგან გამოსაყოფად და ‘?’ ძირითადი მისამართისა და პარამეტრების ერთმანეთისგან გამოსაყოფად

POST გამოიყენება მონაცემების გასაგზავნად სერვერზე, რომლის შენახვა (ან განახლება) უნდა მოხდეს ბაზაში (ან სხვა

რესურში). POST მეთოდი გამოიყენება ფორმებში შეყვანილი ინფორმაციის სერვერზე გასაგზავნად.

POST მეთოდის ელემენტები არ არის ხილვადი, არ ხდება მისი ქეშირება, არ რჩება ბრაუზერის history-ში, მისი

ელემენტების სიგრძე არ არის ლიმიტირებული.

1. რას წარმოადგენს BeatifulSoup მოდულის find() ფუნქცია და რა ტიპის მონაცემს აბრუნებს? მოიყვანეთ მაგალითი.

find() ფუნქცია აბრუნებს მხოლოდ ერთ მნიშვნელობას, პირველივე შემხვედრს.

print(soup.find('ul'))

find\_all() ფუნქცია აბრუნებს პარამეტრად მითითებული ტეგების შიგთავს; თუ ეს ტეგი

რამდენიმეჯერ არის ნახსენები, აბრუნებს ყველას სიის სახით

print(soup.find\_all('ul', {'class':'mylist'}))

1. Flask-ის მოდულის გამოყენებით ვებ აპლიკაციაში როგორ არის შესაძლებელი მოხდეს გვერდის სხვა გვერდზე გადამისამართება. მოიყვანეთ მაგალითი.

url\_for() ფუნქცია უზრუნველყოფს ლინკის დაგენერირებას. პარამეტრად გადაეცემა სტრიქონი,

რომელიც წარმოადგენს იმ ფუნქციის დასახელებას, რომლის შესაბამისი ლინკის აგებაც გვსურს.

ფუნქციას შესაძლოა გადაეცეს სხვა პარამეტრებიც, იმ შემთხვევაში თუ პირველ პარამეტრად

მითითენულ ფუნქცია არის პარამეტრიანი ფუნქცია, ყველა მისი პარამეტრი შეგვიძლია გადავცეთ

url\_for()-ში მომდევნო პარამეტრებად

redirect() ფუნქცია უზრუნველყოფს პარამეტრად გადაცემულ ლინკზე გადასვლას

მაგალითი:

from flask import Flask, redirect, url\_for

app = Flask(\_\_name\_\_)

@app.route('/')

def home():

return 'This is a main page'

@app.route('/admin')

def admin():

return redirect(url\_for('home'))

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

1. რას წარმოადგენს csv მოდული? მოიყვანეთ რომელიმე ფუნქცია შესაბამისი მაგალითით.

Python-ში csv მოდულის გამოყენებით შესაძლებელია csv ფაილიდან მონაცემთა იმპორტი/წაკითხვა

(Read) და/ან მონაცემების ექსპორტი/ჩაწერა (Write) csv ფაილში. csv (Comma Separated Values) ფაილი

გამოიყენება დიდი რაოდენობის მონაცემების შესანახად ტექსტურ ფორმატში. CSV ფაილის გავს

excel-ის ფაილს, სადაც მონაცემები წარმოდგენილია ცხრილის სახით (სტრიქონების და სვეტების

მეშვეობით). csv ფაილში ცხრილის მონაცემები წარმოდგენილია ტექსტურად, რომლის თითოეული

ხაზზე არის ცხრილის თითო სტრიქონის მონაცემი, ხოლო მძიმე გამოიყენება უჯრების (სვეტების)

ერთმანეთისგან გამოსაყოფად.

1. რა მსგავსება და განსხვავებაა პარსინგსა და API-ის შორის.

განსხვავება

პარსინგი არის როდესაც საიტის HTML ს პარსავ და აპი თვითონ საიტის მიერ შექმნილი მონაცემებია, რომელიც კანონის ფარგლებში მოგვაქვს.