Assignment 4(KAZ-EN)

автор: Жақуда Жарқынай

қолданылған бағдарламалар: Python, Flask, HTML, SQLAlchemy

KAZ:

Мәліметтер базасы ‘сұлулық салоны’ қорына мәліметтер қосуға арналған. Қолданушы сервис, күн, уақыт және мастерді таңдай алады. Керекті сервистер код-қа қосылған. Бар істеу керегі сұралынған мәліметтерге жауап беру және ең соңында тоқтатуды ұмытпау:))

ENG:

The database is designed to add data to the "beauty salon" database. The user can choose the service, date, time and even the master. The necessary services are connected to the code.

ашу үшін:

1. pip install virtualenv орнатамыз
2. терминалға virtualenv env енгіземіз
3. активация жасаймыз
4. requirements.txt жүктеп аламыз
5. python app.py ашамыз
6. сервер ақпараттары: localhost, port:5000

to open:

we install pip install virtualenv

we enter virtualenv env in the terminal

we activate

requirements.we download txt

python app.py we open

server information: localhost, port: 5000

from flask import Flask, render\_template, url\_for, request, redirect

from flask\_sqlalchemy import SQLAlchemy

from datetime import datetime

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

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

db = SQLAlchemy(app)

class Todo(db.Model):

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

content = db.Column(db.String(200), nullable=False)

date\_created = db.Column(db.DateTime, default=datetime.utcnow)

def \_\_repr\_\_(self):

return '<Task %r>' % self.id

1. керекті заттарды импорттаймыз және колумдармен жаңа сынып ашамыз

we import the necessary items and open a new class with colums

@app.route('/', methods=['POST', 'GET'])

def index():

if request.method == 'POST':

task\_content = request.form['content']

new\_task = Todo(content=task\_content)

try:

db.session.add(new\_task)

db.session.commit()

return redirect('/')

except:

return 'There was an issue adding your task'

else:

tasks = Todo.query.order\_by(Todo.date\_created).all()

return render\_template('index.html', tasks=tasks)

1. жаңа тапсырма қосуға арналған батырма қосамыз

we add a button for adding a new task

@app.route('/delete/<int:id>')

def delete(id):

task\_to\_delete = Todo.query.get\_or\_404(id)

try:

db.session.delete(task\_to\_delete)

db.session.commit()

return redirect('/')

except:

return 'There was a problem deleting that task'

1. тапсырма өшіруге арналған батырма коды

button code to disable task

@app.route('/update/<int:id>', methods=['GET', 'POST'])

def update(id):

task = Todo.query.get\_or\_404(id)

if request.method == 'POST':

task.content = request.form['content']

try:

db.session.commit()

return redirect('/')

except:

return 'There was an issue updating your task'

else:

return render\_template('update.html', task=task)

1. өңдеуге мүмкіндік беретін батырма

button that allows editing

@app.before\_first\_request

def create\_tables():

db.create\_all()

if \_\_name\_\_ == "\_\_main\_\_":

app.run(debug=True)

1. мәліметтер базасын құрамыз

бұл “app.py” бетіндегі барлық функциялар.

C-create, басты бетте орналасқан

R-read, басты бетте орналасқан

U-update, код жоғарғы жақта

D-delete, код жоғарғы жақта

creating a database

this “app.py " all functions on the surface.

C-create, located on the home page

R-read, located on the home page

U-update, code at the top

D-delete, code at the top

main.css-стильдеуге арналған бет, шрифттар мен түстер орналасқан

main.css-page for styling, where fonts and colors are located

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<link rel="stylesheet" href="{{ url\_for('static', filename='css/main.css') }}">

{% block head %}{% endblock %}

</head>

<body>

{% block body %}{% endblock %}

</body>

</html>

base.html-html бетінің басты денесі

base.html-the main body of an html page

{% extends 'base.html' %}

{% block head %}

<title>Beauty Salon</title>

{% endblock %}

index.html-кесте жасауға арналған бет

index.html-page for creating a table

{% extends 'base.html' %}

{% block head %}

<title>Task Master</title>

{% endblock %}

{% block body %}

<div class="content">

<h1 style="text-align: center">Update Task</h1>

<div class="form">

<form action="/update/{{task.id}}" method="POST">

<input type="text" name="content" id="content" master="content" value="{{task.content}}">

<input type="submit" value="Update">

</form>

</div>

</div>

{% endblock %}

update.html-өңдеуге және өшіруге арналған беттің коды

update.html-page code for editing and erasing

Click==7.0

Flask==1.1.2

Flask-SQLAlchemy==2.4.4

gunicorn==19.9.0

itsdangerous==1.1.0

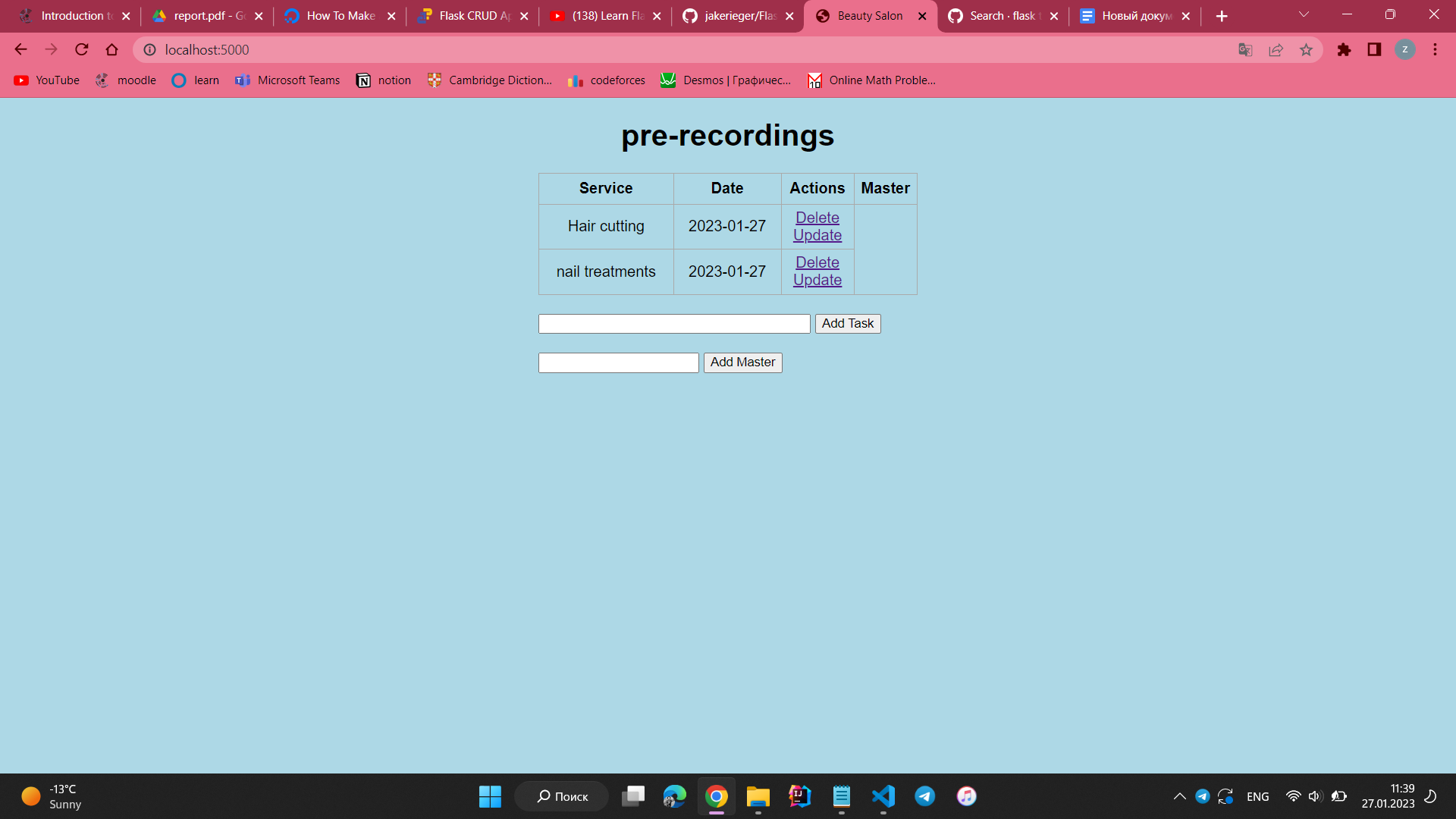
Jinja2==2.11.3

MarkupSafe==1.1.1

SQLAlchemy==1.3.22

Werkzeug==1.0.1

requirements.txt



Файлды run жасаған кезде ашылатын веб-парақша. Осымен болды, қарағаныңызға рахмет!

A web page that opens when the file is created by run. That's it, thank you for watching!