Federal State Budgetary Educational Institution

higher education

Ufa University of Science and Technology

Department of Computational Mathematics and Cybernetics

Laboratory work No.5

“Web Application Development”

Made by: Kozlov I.A.

Group number: PRO 234

Checked by: Rizvanov D.A.

Ufa 2023

**Purpose of the work:** develop a web application for presenting information stored in a database (DB) to the user.

**Task:**

1) Get acquainted with the code and structure of the web application (see sample app Python Flask in the pikpo5\_python\_flask folder). To perform lab work, you can also use the platform as an alternative web framework NodeJS (see sample app NodeJS Express in the pikpo5 folder\_nodejs\_express).

2) Implement methods (queries) for fetching the necessary data from the database for web-applications.

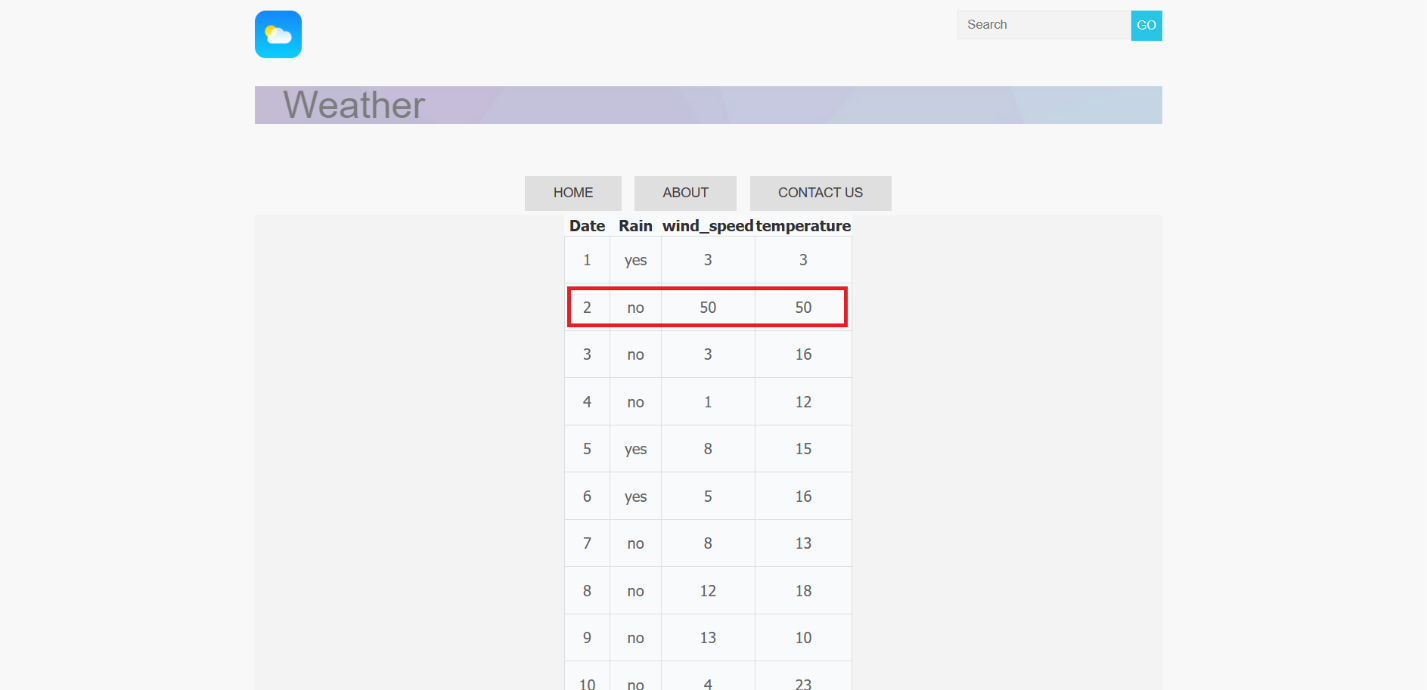
3) Using the built-in web page template engine, implement data insertion into your own templates. html-pages (in accordance with the design of web pages created during laboratory work #4).

4) Implement constant scanning of the specified folder for new files (data sources) in the data handler and their automatic processing.

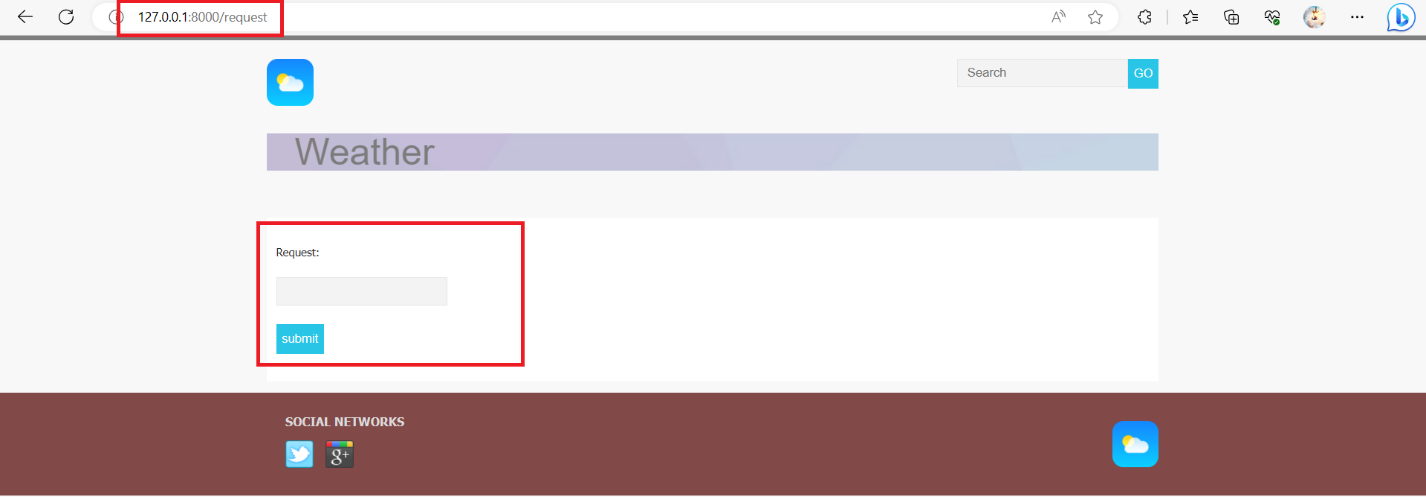
**Description of the work:**

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.

The initial state of website:



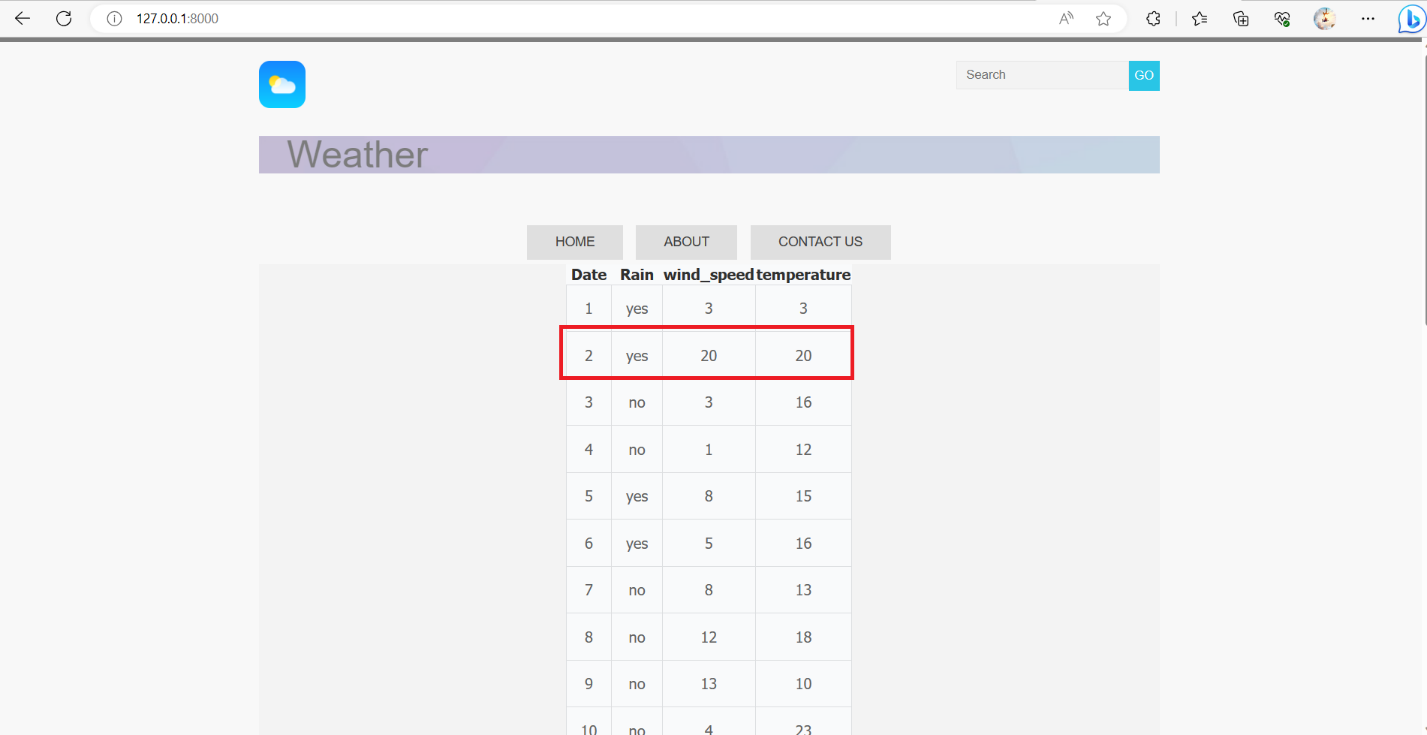
When we forward to /request, we are given functionality to write request.



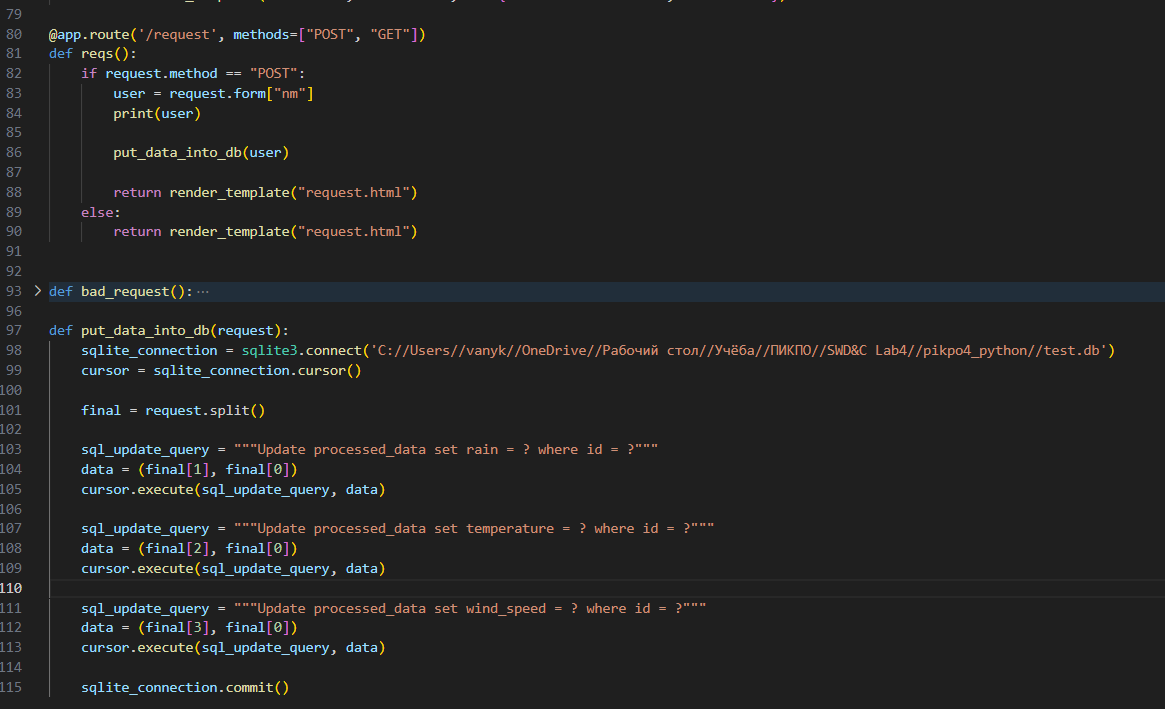
After that we can push to button:



And see the result:



Here you can see how it works:



The route() decorator, @app.route(), binds a URL to a function.

**Conclusion:** I have developed a web application for presenting information stored in a database (DB) to the user.

Github: <https://github.com/Ivan010403/->

Application

Layout.html

<html>

<head>

<meta charset="utf-8">

<title>{{title}}</title>

<link rel="stylesheet" href="/static/stylesheets/styles.css">

<script type="text/javascript" src="/static/javascripts/jquery-3.6.0.min.js"></script>

</head>

<body>

<div class="wrapper">

<header>

<a href="/"><img src="/static/images/logo2.png" height="50" weight ="50" alt="Whitesquare logo"></a>

<form name="search" class="sinput" action="#" method="get">

<input type="search" name="q" placeholder="Search"><button type="submit">GO</button>

</form>

<div class="heading">

<h1 id="head\_name">Weather</h1>

</div>

</header>

{% if navmenu %}

<nav>

<ul class="top-menu">

{% for item in navmenu %}

<li><a href="{{item.addr}}">{{item.name}}</a></li>

{% endfor %}

</ul>

</nav>

{% endif %}

<main id="{{page\_name}}">

{% block content %}{% endblock %}

</main>

</div>

<footer>

<div class="footer">

<div id="social">

<h3>SOCIAL NETWORKS</h3>

<a href="http://twitter.com/" class="social-icon twitter"></a>

<a href="http://google.com/" class="social-icon google"></a>

</div>

<div id="footer-logo">

<a href="/"><img src="/static/images/logo2.png" height="50" weight ="50" alt="Whitesquare logo"></a>

</div>

</div>

</footer>

<script type="text/javascript" src="/static/javascripts/topmenu.js"></script>

<script type="text/javascript" src="/static/javascripts/jqfunc.js"></script>

</body>

</html>

Request.html

{% extends "layout.html" %}

{% block title %} request {% endblock %}

{% block content %}

<form action="#" method="post">

<p>Request:</p>

<p><input type="text" name="nm" /></p>

<p><input type="submit" value="submit" /></p>

</form>

{% endblock %}

Router.py

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

def reqs():

if request.method == "POST":

user = request.form["nm"]

print(user)

put\_data\_into\_db(user)

return render\_template("request.html")

else:

return render\_template("request.html")

def put\_data\_into\_db(request):

sqlite\_connection = sqlite3.connect('C://Users//vanyk//OneDrive//Рабочий стол//Учёба//ПИКПО//SWD&C Lab4//pikpo4\_python//test.db')

cursor = sqlite\_connection.cursor()

final = request.split()

sql\_update\_query = """Update processed\_data set rain = ? where id = ?"""

data = (final[1], final[0])

cursor.execute(sql\_update\_query, data)

sql\_update\_query = """Update processed\_data set temperature = ? where id = ?"""

data = (final[2], final[0])

cursor.execute(sql\_update\_query, data)

sql\_update\_query = """Update processed\_data set wind\_speed = ? where id = ?"""

data = (final[3], final[0])

cursor.execute(sql\_update\_query, data)

sqlite\_connection.commit()