

ГУАП

КАФЕДРА № 43

ОТЧЕТ  
ЗАЩИЩЕН С ОЦЕНКОЙ  
ПРЕПОДАВАТЕЛЬ

Старший преподаватель

должность, уч. степень,  
звание

Соловьева Н.А.

подпись, дата

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №6  
**Серверные сценарии. Работа с базой данных.**

по дисциплине: Web-Технологии

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР.

4134к

подпись, дата

Столяров Н.С.

инициалы, фамилия

Санкт-Петербург  
2024

**Цель работы:** изучение трехуровневой архитектуры веб-приложений (Клиент – Веб-сервер – Сервер БД) и языка написания серверного сценария php для построения динамического контента web-сайтов.

## Базовое задание

Установить компоненты 3-х уровневой архитектуры. В качестве звеньев 3-х уровневой архитектуры разрешается использовать Apache+PHP+MySQL (пакет программ WAMP – XAMPP, Денвер и т.п.). Язык серверных сценариев - PHP.

Подготовить базу данных, состоящую из 2 таблиц, связанные между собой отношением «один-ко-многим». Каждая таблица должна находиться в 3 нормальной форме и содержать не менее 5 полей. Содержимое полей таблицы должно соответствовать теме сайта. Все данные из таблиц должны отображаться на странице, включенной в разрабатываемый сайт. Для формирования веб-страницы использовать команду echo.

## Расширенное задание

- 1.. не отображать служебные поля (первичные и вторичные ключи). +
- 2.. все поля отобразить в одной таблице на веб-странице +
- 3.. добавить возможность ввода новой строки в таблицу базы на веб-странице (+ частично тип сома форма)
- 4.. добавить возможность редактировать строку в таблицу базы на веб-странице +
- 5.. добавить возможность удаления строки из таблицы базы на веб-странице +
- 6.. написать серверный сценарий на php, выполняющий задание с матрицами из работы № 3.
- 7.. применить оформление через css +
- 8.. несколько раз для формирования веб-страницы использовать функцию printf()

## Листинг Скрипта

Main.html

```
<html>
<head>
  <meta charset="utf-8">
  <title>REGISTER</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="icon" href="/static/img/favicon.svg" type="image/svg+xml">
  <link rel="stylesheet" href="static/css/main.css"/>
  <link rel="stylesheet" href="static/css/form.css"/>
  <link rel="stylesheet" href="static/css/checkbox.css"/>
  <link href='http://fonts.googleapis.com/css?family=Roboto' rel='stylesheet' type='text/css'>
</head>
<body>
  <div id="form" class="form center_width">
    <h1 id="server_warning" align="center" style="display: none">Сервис временно
недоступен.</h1>
    <div id="first" style="display: block">
      <input id="last_name" type="text" class="input_style input_form"
placeholder="Фамилия">
```

```
<br>
<input id="name" type="text" class="input_style input_form" placeholder="Имя">
<br>
<input id="surname" type="text" class="input_style input_form"
placeholder="Отчество">
<br>
<input id="phone" type="text" class="input_style input_form" placeholder="Телефон"
onfocus="on_focus('phone', 'phone')" onblur="on_blur('phone', 'phone')">
<br>
<select id="city" name="city" class="input_style input_form">
  <option value="" selected>Выберете город</option>
  <option value="Москва">Москва</option>
  <option value="Санкт-Петербург">Санкт-Петербург</option>
</select>
<select id="sex" name="sex" class="input_style input_form">
  <option value="" selected>Ваш пол</option>
  <option value="male">Мужской</option>
  <option value="female">Женский</option>
</select>
<br>
<br>
<p class="form_comment">Дата рождения</p>
<input id="date" type="date" class="input_style input_form">
<br>
<br>
<p class="form_comment">Лицевая сторона паспорта</p>
<label for="front_pasport">
  <div class="input_file block_select" id="front_pasport_div">
    <div>
      <p id="front_pasport_name">Загрузить файл</p>
    </div>
    <div class="loaded_file" id="front_pasport_loaded">
      
    </div>
  </div>
</label>
<br>
<p class="form_comment">Страница паспорта с регистрацией</p>
<label for="register_pasport">
  <div class="input_file block_select" id="register_pasport_div">
    <div>
      <p id="register_pasport_name">Загрузить файл</p>
    </div>
    <div class="loaded_file" id="register_pasport_loaded">
      
    </div>
  </div>
</label>
<br>
<p class="form_comment">Водительское удостоверение лицевая сторона</p>
<label for="front_certificate">
  <div class="input_file block_select" id="front_certificate_div">
```

```

<div>
  <p id="front_certificate_name">Загрузить файл</p>
</div>
<div class="loaded_file" id="front_certificate_loaded">
  
</div>
</div>
</label>
<br>
<p class="form_comment">Водительское удостоверение обратная сторона</p>
<label for="back_certificate">
  <div class="input_file block_select" id="back_certificate_div">
    <div>
      <p id="back_certificate_name">Загрузить файл</p>
    </div>
    <div class="loaded_file" id="back_certificate_loaded">
      
    </div>
  </div>
</label>
  <input id="snils" type="text" class="input_style input_form" placeholder="Введите номер СНИЛС" onfocus="on_focus('snils', 'snils')" onblur="on_blur('snils', 'snils')" style="width: 100%">
  <input id="inn" type="text" class="input_style input_form" placeholder="Введите ИНН" onfocus="on_focus('inn', 'inn')" onblur="on_blur('inn', 'inn')" style="width: 100%">
  <div style="margin: 0px">
    <input type="checkbox" class="custom-checkbox" id="checkbox_policy" name="checkbox_policy" value="yes">
    <label for="checkbox_policy" style="margin-top: 0px;"><p>Я согласен с <a href="{{ policy }}" target="_blank">политикой конфиденциальности</a></p></label>
  </div>
  <div id="phone_code_div" style="display: none">
    <input id="phone_code" type="text" class="input_style input_form" placeholder="Код подтверждения" onfocus="on_focus('phone_code', 'phone_code')" onblur="on_blur('phone_code', 'phone_code')" style="width: 100%">
    <br>
    <br>
  </div>
  <div class="send_button center_width block_select" onclick="next()">
    <p id="send_button_text">Отправить</p>
    <div id="send_button_progress" class="send_button_progress"></div>
  </div>
  <!-- <div class="send_button center_width block_select" onclick="test_event()">
    <p id="send_button_text">test</p>
    <div id="send_button_progress" class="send_button_progress"></div>
  </div> -->
  <br>
  <br>
</div>
<div id="end" class="end" style="display: none">
  <h1 align="center">Заявка отправлена</h1>
</div>

```

```

</div>
<input type="file" id="front_pasport" name="front_pasport" style="display: none"
onchange="selected_file('front_pasport')" accept=".jpg, .png, .jpeg, .gif, .bmp, .tif,
.tiff|image/*">
<input type="file" id="register_pasport" name="register_pasport" style="display: none"
onchange="selected_file('register_pasport')" accept=".jpg, .png, .jpeg, .gif, .bmp, .tif, .tiff|
image/*">
<input type="file" id="front_certificate" name="front_certificate" style="display: none"
onchange="selected_file('front_certificate')" accept=".jpg, .png, .jpeg, .gif, .bmp, .tif, .tiff|
image/*">
<input type="file" id="back_certificate" name="back_certificate" style="display: none"
onchange="selected_file('back_certificate')" accept=".jpg, .png, .jpeg, .gif, .bmp, .tif, .tiff|
image/*">
</body>
<script type="text/javascript" src="static/js/main.js"></script>
<script type="text/javascript" src="static/js/form.js"></script>
</html>

```

## Reg.py

```

from fastapi import UploadFile
import requests
import json
import deco

from globals import *
from ping import ping
from log import log

@deco.try_decorator
async def send_register(
    name: str,
    last_name: str,
    surname: str,
    date: str,
    snils: str,
    inn: str,
    phone: str,
    city: str,
    sex: str,
    front_pasport: UploadFile,
    register_pasport: UploadFile,
    front_certificate: UploadFile,
    back_certificate: UploadFile
):
    if not ping(ONEC_HOST_CANDIDATE):
        return {
            "Success": False
        }

# заходим в нужные разделы БД

```

```

database = MONGODB_CLIENT[MONGODB_DATABASE]
collection = database[MONGODB_COLLECTION]

# данные которые надо записать в БД
data = {
    "name": name,          # Имя
    "last_name": last_name, # Фамилия
    "surname": surname,    # Отчество
    "sex": sex,            # Пол
    "birth_date": date,    # Дата рождения
    "snils": snils,        # СНИЛС
    "inn": inn,            # ИНН
    "phone": phone,        # Номер телефона
    "city": city           # Номер телефона
}

# сохраняем данные в БД и получаем индикатор
inserted_id = collection.insert_one(data).inserted_id

# загружаем файлы в minio
MINIO_CLIENT.put_object(MINIO_BUCKET, f'{inserted_id}/front_pasport.
{front_pasport.filename.split('.')[-1]}', front_pasport.file, length=front_pasport.size)
MINIO_CLIENT.put_object(MINIO_BUCKET, f'{inserted_id}/register_pasport.
{register_pasport.filename.split('.')[-1]}', register_pasport.file, length=register_pasport.size)
MINIO_CLIENT.put_object(MINIO_BUCKET, f'{inserted_id}/front_certificate.
{front_certificate.filename.split('.')[-1]}', front_certificate.file, length=front_certificate.size)
MINIO_CLIENT.put_object(MINIO_BUCKET, f'{inserted_id}/back_certificate.
{back_certificate.filename.split('.')[-1]}', back_certificate.file, length=back_certificate.size)

# MINIO_CLIENT.fput_object(MINIO_BUCKET, f'{inserted_id}/front_pasport.
{front_pasport.filename.split('.')[-1]}', front_pasport.file.fileno())
# MINIO_CLIENT.fput_object(MINIO_BUCKET, f'{inserted_id}/register_pasport.
{register_pasport.filename.split('.')[-1]}', register_pasport.file.fileno())
# MINIO_CLIENT.fput_object(MINIO_BUCKET, f'{inserted_id}/front_certificate.
{front_certificate.filename.split('.')[-1]}', front_certificate.file.fileno())
# MINIO_CLIENT.fput_object(MINIO_BUCKET, f'{inserted_id}/back_certificate.
{back_certificate.filename.split('.')[-1]}', back_certificate.file.fileno())

data = {
    "id": str(inserted_id), # ID (через него можно получить изображения с MinIO)
    "name": name,          # Имя
    "last_name": last_name, # Фамилия
    "surname": surname,    # Отчество
    "sex": sex,            # Пол
    "birth_date": date,    # Дата рождения
    "snils": snils,        # СНИЛС
    "inn": inn,            # ИНН
    "phone": phone,        # Номер телефона
    "city": city           # Номер телефона
}

# Изображения документов лежат в бакете registertest

```

```

# путь к изображениям:
# {inserted_id}/front_pasport.{type}
# {inserted_id}/register_pasport.{type}
# {inserted_id}/front_certificate.{type}
# {inserted_id}/back_certificate.{type}

headers = {"Content-Type": "application/json; charset=utf-8"}
# print(data)
json_data = json.dumps(data)

log.print(f'Send candidate with id {str(inserted_id)}', "INFO ")

r = requests.post(ONEC_HOST_CANDIDATE, data=json_data, headers=headers)
r.encoding = 'utf-8'
# print(r.text)

return {"inserted_id": str(inserted_id), "Success": True}

```

sms.py

```

from fastapi.responses import RedirectResponse
import starlette.status as status
from fastapi import *
import deco
import json
import requests

from globals import *
from ping import ping
from log import log

@deco.try_decorator
async def send_sms(phone:str):
    if not ping(ONEC_HOST + "auth"):
        return {
            "Success": False
        }

    headers = {"Content-Type": "application/json; charset=utf-8"}

    data = {
        "request": "sendAnySMS",
        "PhoneNumber": str(phone)
    }

    json_data = json.dumps(data)

    log.print(f'Send sms to {phone}', "INFO ")

    r = requests.post(ONEC_HOST + "auth", data=json_data, headers=headers)
    r.encoding = 'utf-8'
    response = r.json()

```

return response

## Скриншоты

Заполните информацию о себе — это поможет быстрее подготовить документы для трудоустройства

Фамилия
Имя
Отчество
Телефон
Выберете город
Ваш пол
Дата рождения
Лицевая сторона паспорта
Страница паспорта с регистрацией
Водительское удостоверение лицевая сторона
Водительское удостоверение обратная сторона
Введите номер СНИЛС
Введите ИНН
<input type="checkbox"/> Я согласен с <a href="#">политикой конфиденциальности</a>
Отправить