Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Уфимский университет науки и технологий»

Кафедра ВВТиС

**ОТЧЕТ**

**по учебной практике**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Группа ПМ-355 | Фамилия И.О. | Подпись | Дата | Оценка |
| Студент | Ахтамьянов Р.Р. |  |  |  |
| Руководитель  от базы практики | Ямилева А.М. |  |  |  |
| Руководитель от университета | Ямилева А.М. |  |  |  |

Уфа 2023

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Уфимский университет науки и технологий»

Кафедра высокопроизводительных вычислительных технологий и систем

**ЗАДАНИЕ**

на учебную практику

Студент: Ахтамьянов Роберт Рустемович Группа: ПМ-355

Период практики: 2.06.2023 – 4.07.2023

База практики: каф. ВВТиС ФГБОУ ВО УНИИТ

Руководитель от базы практики: Ямилева А.М.

Руководитель от университета: Ямилева А.М.

Тема практики

Разработка веб приложения на фреймворке Flask.

Задачи исследования

1. Изучить принципы работы bootstrap и html. Разработать веб-приложение на фреймворке Flask.

**Список рекомендуемой литературы**

1 Калакуцкий, А. Введение в Python. Практикум : образовательный курс / Образовательная платформа Stepik. — Режим доступа:

<https://stepik.org/course/56391/syllabus>. — Загл. с экрана.

2. Задойный, А. Практикум по математике и Python : образовательный курс / Образовательная платформа Stepik. — Режим доступа:

3. <https://stepik.org/course/3356/syllabus>. — Загл. с экрана.

4. Гринберг. Разработка веб-приложений с использованием Flask на языке Python

|  |  |
| --- | --- |
| Дата выдачи задания  2 июня 2023 г. | Дата окончания работы  4 июля 2023 г. |

Руководитель от базы практики \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ямилева А.М.

Руководитель от университета \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ямилева А.М.

**СОДЕРЖАНИЕ**

[Введение 5](#_Toc77280741)

[1 Теоретическая часть 6](#_Toc77280742)

[2 Практическая часть 9](#_Toc77280743)

[2.1 Язык программирования Python 9](#_Toc77280744)

[2.2 Web приложение 15](#_Toc77280747)

[2.2.1 Индивидуальное задание 15](#_Toc77280748)

[Заключение 18](#_Toc77280749)

[Список литературы 19](#_Toc77280750)

[Приложение А 31](#_Toc77280752)

# Введение

Python – это высокоуровневый язык программирования, который имеет множество различных способов применений, включая науку о данных и внутреннюю веб-разработку. Он является мощным инструментом для анализа данных, широко используется в технологии больших данных.

В данной работе в качестве индивидуального задания рассматривается создание своего собственного сайта с использованием bootstrap, регистрацией пользователей, OMP.

Цель – используя полученные знания о языке Python, разработать свой сайт.

Задачи исследования:

* Изучить основы языка программирования Python и теоретический материал в области разработки web приложений.
* Реализовать регистрацию и авторизацию пользователей.
* Реализовать полученный алгоритм на языке Python, Html.

# 1 Теоретическая часть

1.1 Язык программирования Python

Python — это скриптовый язык программирования. Он универсален, поэтому подходит для решения разнообразных задач и многих платформ, начиная с iOS и Android и заканчивая серверными ОС.

Это интерпретируемый язык — он не компилируется, то есть до запуска представляет из себя обычный текстовый файл. Программировать можно практически на всех платформах, язык хорошо спроектирован и логичен.

1.1.4 Flask( веб-фреймворк)

Flask — фреймворк для создания веб-приложений на языке программирования Python, использующий набор инструментов Werkzeug, а также шаблонизатор Jinja2. Относится к категории так называемых микрофреймворков[en] — минималистичных каркасов веб-приложений, сознательно предоставляющих лишь самые базовые возможности.

# 2 Практическая часть

## 2.1 Язык программирования Python

### 2.1.1 Основы Python, библиотеки Numpy

В этом разделе в рамках курса с сайта stepik.org были получены базовые навыки написания программ на языке Python, изучены управляющие структуры в Python, строки и списки, функции, коллекции и генераторы, работа с файлами и с библиотекой NumPy.

## 2.2 Web приложение

Учебная практика состоит из двух частей:

* Изучение языка программирования Python, а также его библиотек: numpy, и их применение при моделировании. Необходимо прослушать лекции и выполнить ряд практических задач.
* Индивидуальное задание от научного руководителя.

### 2.2.1 Индивидуальное задание

Необходимо реализовать веб-приложение на фреймворке flask со следующими элементами:

* Регистрация и авторизация пользователей, разное содержимое страниц в зависимости от авторизации, защита приватных страниц.
* Работа с базой данных кафедры.

Листинг программы представлен в приложении А. Результат работы программы предоставлен на рисунках ниже.

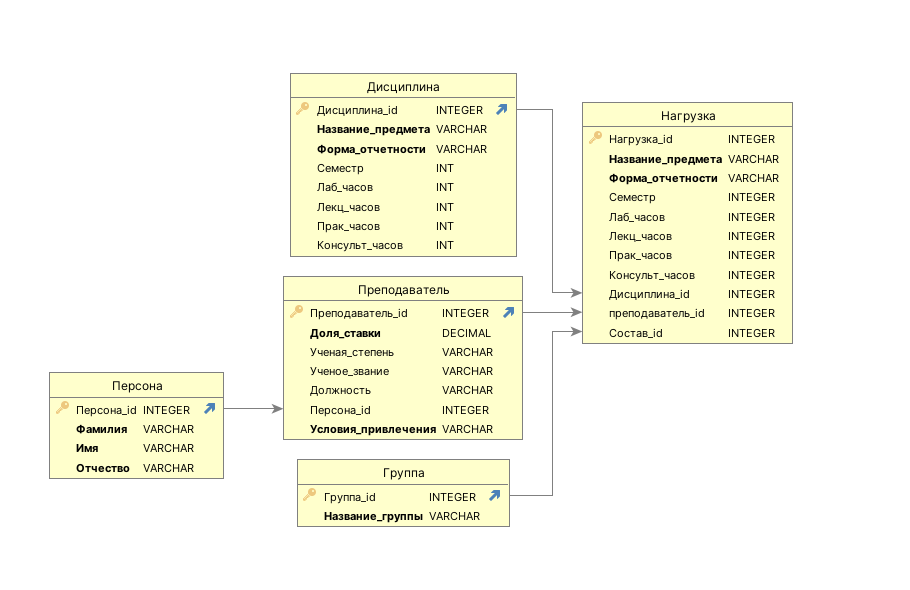


Рисунок 1– База данных

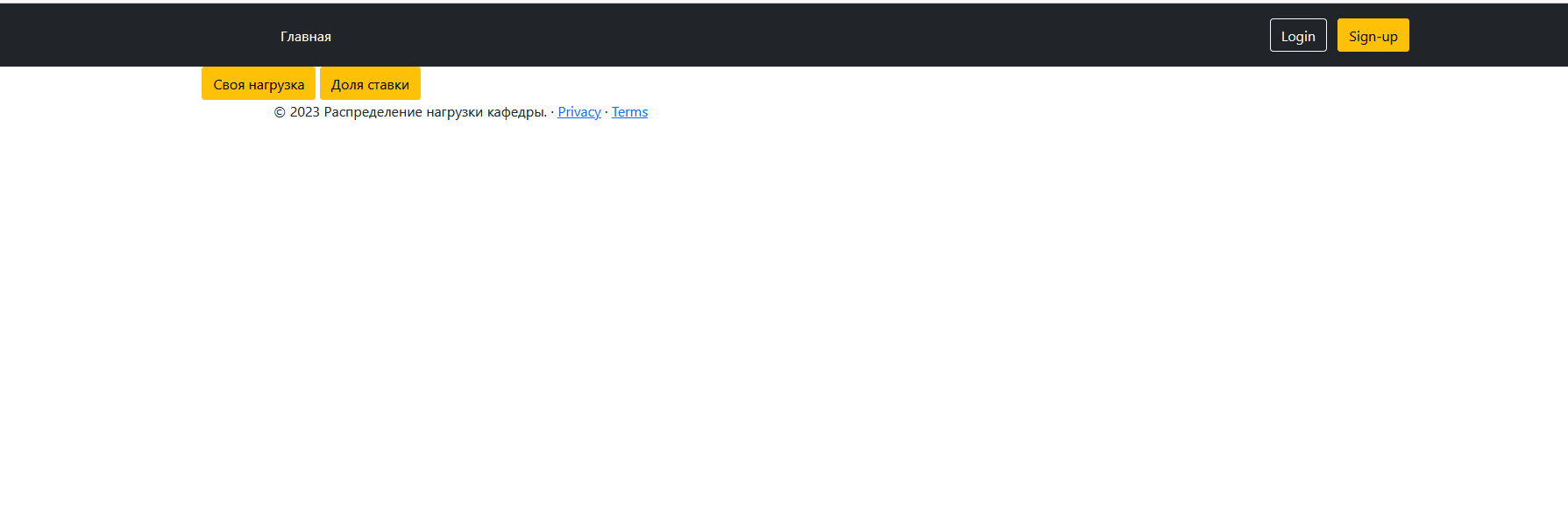


Рисунок 2 – Страница для преподавателя

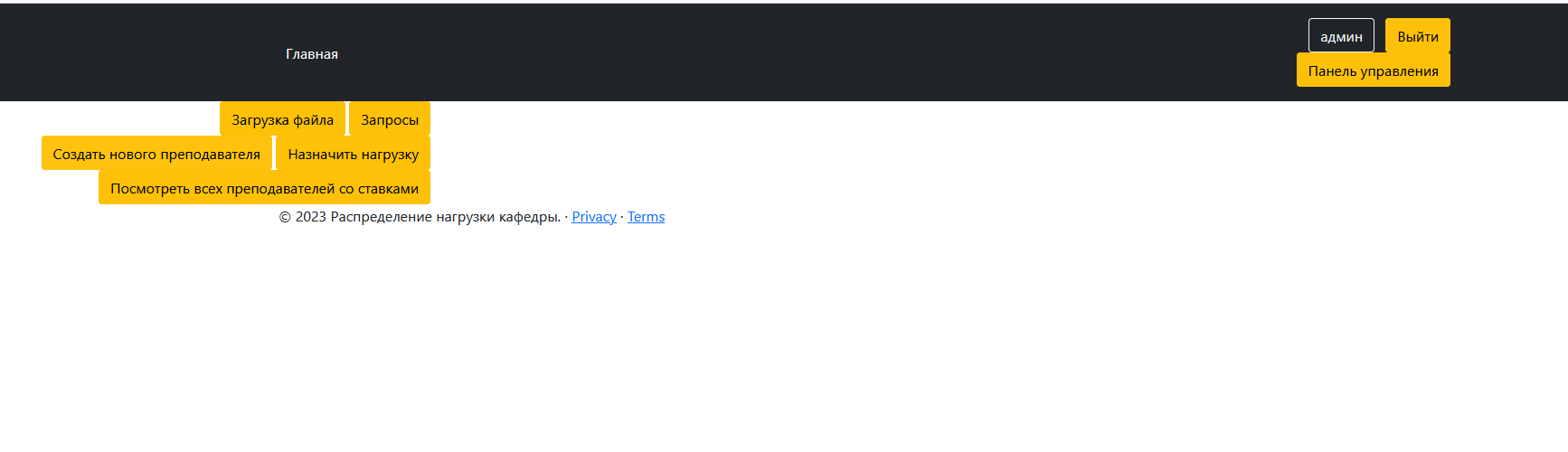
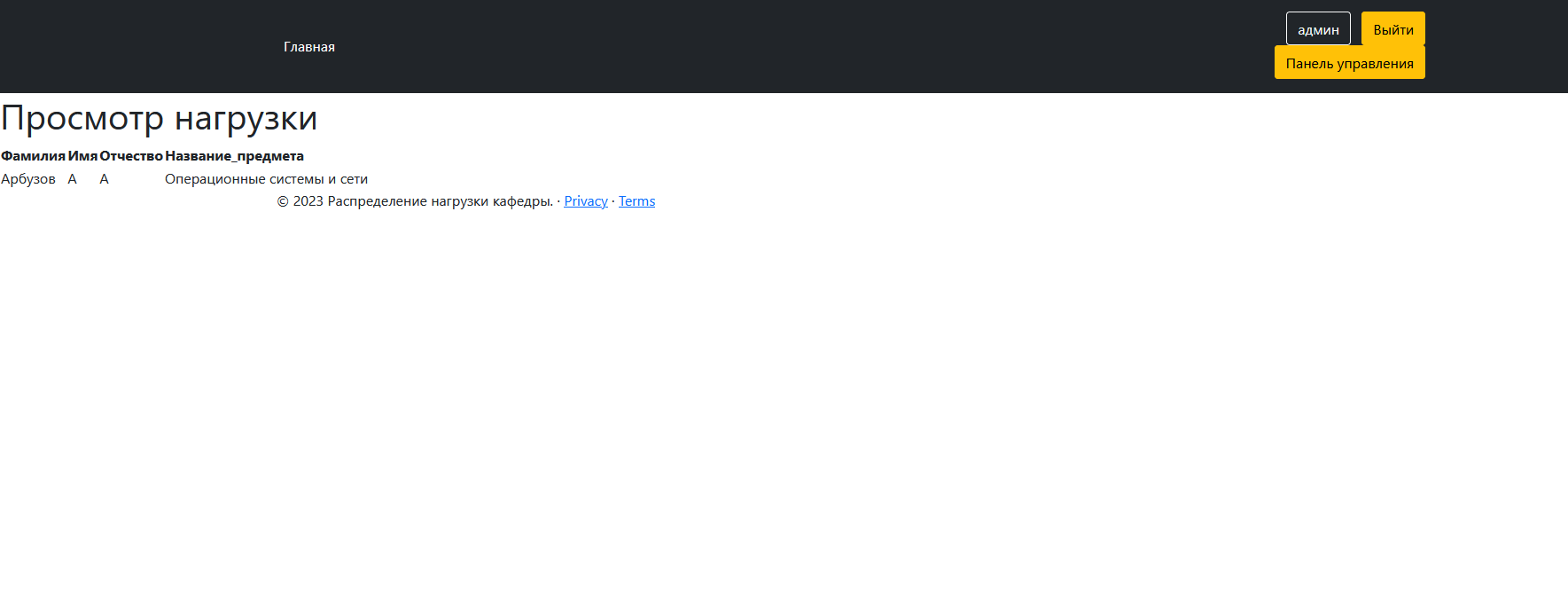
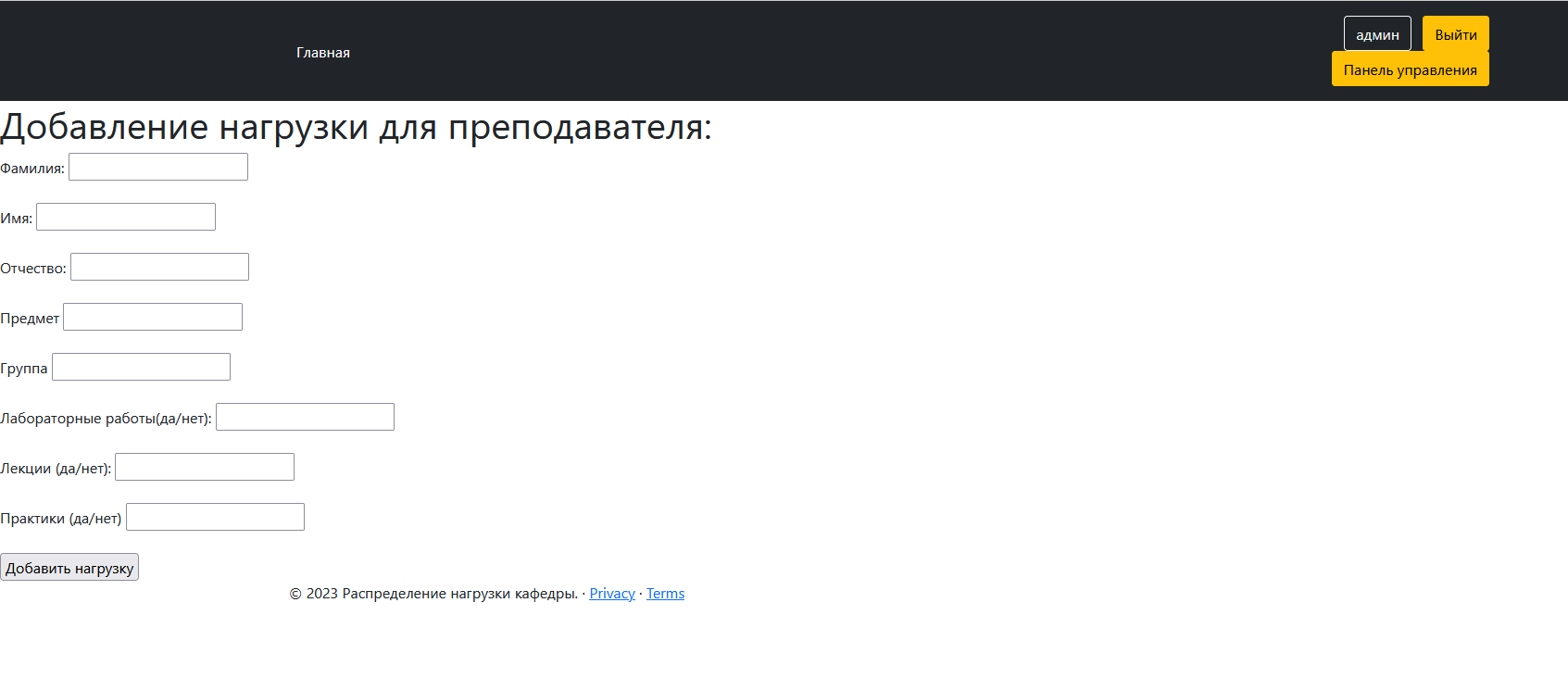
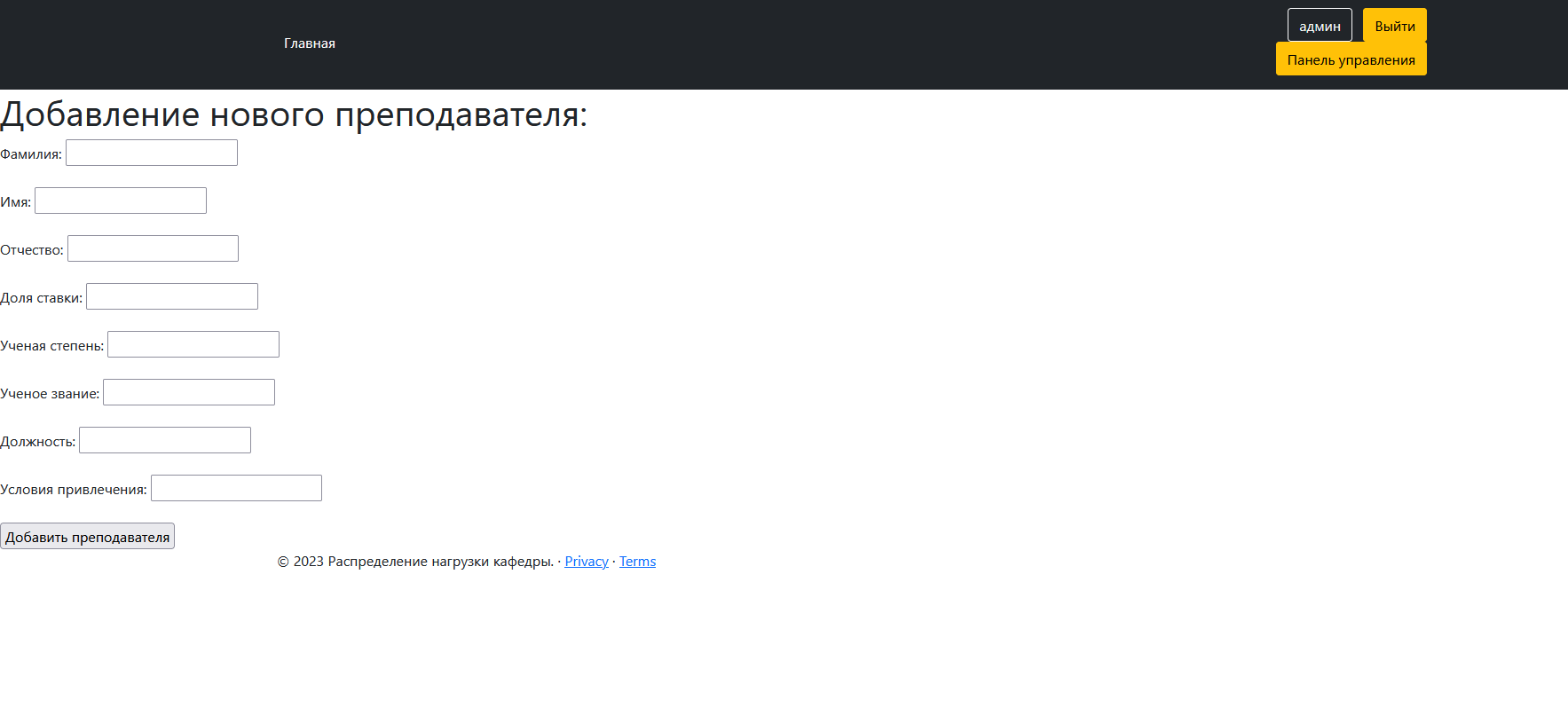


Рисунок 5 – Страница для администратора







# Заключение

В результате учебной практики были приобретены новые теоретические и практические знания, навыки и умения. В ходе практики был изучен язык программирования Python. Он оказался прост в применении и понимании, что облегчило выполнение индивидуального задания.

В ходе летней учебной практики были выполнены следующие задачи:

* Ознакомление со средой разработки
* Изучение базовых возможностей языка программирования Python.
* Изучение основ создания веб-интерфейса и фреймворка Flask.
* Реализация веб-приложения.

# Список литературы

1. Калакуцкий, А. Введение в Python. Практикум : образовательный курс / Образовательная платформа Stepik. — Режим доступа:

<https://stepik.org/course/56391/syllabus>. — Загл. с экрана.

1. Задойный, А. Практикум по математике и Python : образовательный курс / Образовательная платформа Stepik. — Режим доступа:

<https://stepik.org/course/3356/syllabus>. — Загл. с экрана.

1. ITC. Веб-разработка для начинающих: HTML и CSS : образовательный курс / Образовательная платформа Stepik. — Режим доступа:

https://stepik.org/course/38218/syllabus. — Загл. с экрана.

# Приложение Web

**Main.py**

from flask import Flask, render\_template, url\_for, redirect  
from data import db\_session  
from data.users import User  
from data.registerForm import RegisterForm  
from data.loginForm import LoginForm  
from flask\_login import LoginManager, login\_user, logout\_user, login\_required  
  
  
  
  
db\_session.global\_init("db/blogs.sqlite")  
  
app = Flask(\_\_name\_\_)  
app.config['SECRET\_KEY'] = 'Admin123'  
login\_manager = LoginManager()  
login\_manager.init\_app(app)  
  
  
@login\_manager.user\_loader  
def load\_user(user\_id):  
 session = db\_session.create\_session()  
 return session.query(User).get(user\_id)  
  
  
@app.route('/logout')  
@login\_required  
def logout():  
 logout\_user()  
 return redirect("/")  
  
  
@app.route("/")  
@app.route('/home')  
def home():  
 return render\_template('home.html')  
  
  
  
def paz():  
 return render\_template('paz.html')  
  
  
@app.route('/order')  
def order():  
 return render\_template('order.html')  
  
  
@app.route('/login', methods=['GET', 'POST'])  
def login():  
 form = LoginForm()  
 if form.validate\_on\_submit():  
 session = db\_session.create\_session()  
 user = session.query(User).filter(User.email == form.email.data).first()  
 if user and user.check\_password(form.password.data):  
 login\_user(user, remember=form.remember\_me.data)  
 return redirect("/")  
 return render\_template('login.html', message="Неправильный логин или пароль", form=form)  
 return render\_template('login.html', title='Авторизация', form=form)  
  
  
@app.route('/register', methods=['GET', 'POST'])  
def reqister():  
 form = RegisterForm()  
 if form.validate\_on\_submit():  
 if form.password.data != form.password\_again.data:  
 return render\_template('register.html', title='Регистрация',form=form, message="Пароли не совпадают")  
 session = db\_session.create\_session()  
 if session.query(User).filter(User.email == form.email.data).first():  
 return render\_template('register.html', title='Регистрация', form=form, message="Такой пользователь уже есть")  
 user = User(name=form.name.data, email=form.email.data, about=form.about.data)  
 user.set\_password(form.password.data)  
 session.add(user)  
 session.commit()  
 return redirect('/login')  
 return render\_template('register.html', title='Регистрация', form=form)  
  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 db\_session.global\_init("db/blogs.sqlite")  
 #table\_vaz()  
 #table\_gaz()  
 #table\_yaz()  
 #table\_paz()  
 app.run()

\_\_all\_models.py

from . import users

db\_session.py

import sqlalchemy as sa  
import sqlalchemy.orm as orm  
from sqlalchemy.orm import Session  
import sqlalchemy.ext.declarative as dec  
  
  
SqlAlchemyBase = dec.declarative\_base()  
  
  
\_\_factory = None  
  
  
def global\_init(db\_file):  
 global \_\_factory  
  
 if \_\_factory:  
 return  
 if not db\_file or not db\_file.strip():  
 raise Exception("Необходимо указать файл базы данных.")  
  
 conn\_str = f'sqlite:///{db\_file.strip()}?check\_same\_thread=False'  
 print(f"Подключение к базе данных по адресу {conn\_str}")  
  
 engine = sa.create\_engine(conn\_str, echo=False)  
 \_\_factory = orm.sessionmaker(bind=engine)  
  
 from . import \_\_all\_models  
  
 SqlAlchemyBase.metadata.create\_all(engine)  
  
  
def create\_session() -> Session:  
 global \_\_factory  
 return \_\_factory()

loginForm.py

from flask\_wtf import FlaskForm  
from wtforms import StringField, PasswordField, SubmitField, BooleanField  
from wtforms.validators import DataRequired  
  
class LoginForm(FlaskForm):  
 email = StringField('Почта', validators=[DataRequired()])  
 password = PasswordField('Пароль', validators=[DataRequired()])  
 remember\_me = BooleanField('Запомнить меня')  
 submit = SubmitField('Войти')

registerForm.py

from flask\_wtf import FlaskForm  
from wtforms import StringField, PasswordField, SubmitField, TextAreaField  
from wtforms.validators import DataRequired  
  
class RegisterForm(FlaskForm):  
 email = StringField('Почта', validators=[DataRequired()])  
 password = PasswordField('Пароль', validators=[DataRequired()])  
 password\_again = PasswordField('Повторите пароль', validators=[DataRequired()])  
 name = StringField('Имя пользователя', validators=[DataRequired()])  
 about = TextAreaField("Немного о себе")  
 submit = SubmitField('Войти')

users.py

import datetime  
import sqlalchemy  
from .db\_session import SqlAlchemyBase  
from flask\_login import UserMixin  
from werkzeug.security import generate\_password\_hash, check\_password\_hash  
  
class User(SqlAlchemyBase, UserMixin):  
 \_\_tablename\_\_ = 'users'  
  
 id = sqlalchemy.Column(sqlalchemy.Integer, primary\_key=True, autoincrement=True)  
  
 name = sqlalchemy.Column(sqlalchemy.String, nullable=True)  
 about = sqlalchemy.Column(sqlalchemy.String, nullable=True)  
 email = sqlalchemy.Column(sqlalchemy.String, index=True, unique=True, nullable=True)  
  
 hashed\_password = sqlalchemy.Column(sqlalchemy.String, nullable=True)  
 created\_date = sqlalchemy.Column(sqlalchemy.DateTime, default=datetime.datetime.now)  
  
  
 def set\_password(self, password):  
 self.hashed\_password = generate\_password\_hash(password)  
  
 def check\_password(self, password):  
 return check\_password\_hash(self.hashed\_password, password)

base.html

<!doctype html>  
<html lang="ru">  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport"  
 content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">  
 <meta http-equiv="X-UA-Compatible" content="ie=edge">  
  
  
 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">  
 <title>{% block title %}{% endblock %}</title>  
 <style>  
 #body { background: url(https://st3.depositphotos.com/4035529/13152/v/600/depositphotos\_131522432-stock-illustration-auto-spare-parts.jpg); }  
 .ttt img  
 {  
 border: 2px solid #55c5e9; /\* Рамка вокруг фотографии \*/  
 padding: 10px; /\* Расстояние от картинки до рамки \*/  
 background: #666; /\* Цвет фона \*/  
 margin-right: 5px; /\* Отступ справа \*/  
 margin-bottom: 5px; /\* Отступ снизу \*/  
 }  
 </style>  
  
</head>  
<body>  
</div>  
  
  
  
 <header class="p-3 bg-dark text-white">  
 <div class="container">  
 <div class="d-flex flex-wrap align-items-center justify-content-center justify-content-lg-start">  
  
 <ul class="nav col-12 col-lg-auto me-lg-auto mb-2 justify-content-center mb-md-0">  
 <li><a href="/home" class="nav-link px-2 text-white">Главная</a></li>  
 <li><a href="/vaz" class="nav-link px-2 text-white">Ваз</a></li>  
 <li><a href="/gaz" class="nav-link px-2 text-white">Газ</a></li>  
 <li><a href="/yaz" class="nav-link px-2 text-white">Уаз</a></li>  
 <li><a href="/paz" class="nav-link px-2 text-white">Паз</a></li>  
 <li><a href="/order" class="nav-link px-2 text-white">Заказ</a></li>  
 {% if current\_user.is\_authenticated %}  
 <li><a class="nav-link px-2 text-white"> Ваш купон на скидку:</a></li>  
 <li><a class="nav-link px-2 text-black"> Avto3</a></li>  
 {% else %}  
 <li><a class="nav-link px-2 text-red">Зарегестрируйтесь и получите купон на скидку в 3 процента!</a></li>  
 {% endif %}  
  
 </ul>  
  
  
 {% if current\_user.is\_authenticated %}  
 <div class="col-md-3 text-end">  
 <button type="button" class="btn btn-outline-light me-2" onClick="location.href='#'">{{ current\_user.name }}</button>  
 <button type="button" class="btn btn-warning" onClick="location.href='/logout'">Выйти</button>  
 </div>  
 {% else %}  
 <div class="col-md-3 text-end">  
 <button type="button" class="btn btn-outline-light me-2" onClick="location.href='/login'">Login</button>  
 <button type="button" class="btn btn-warning" onClick="location.href='/register'">Sign-up</button>  
 </div>  
 </div>  
 {% endif %}  
  
 </div>  
 </div>  
 </header>  
  
  
 {% block body %}{% endblock %}  
 <div class="container">  
 <p>© 2021 Автодеталь. · <a href="#">Privacy</a> · <a href="#">Terms</a></p>  
 </div>  
  
</body>  
</html>

home.html

{% extends 'base.html' %}  
<head>  
 <meta charset="UTF-8">  
  
  
 {% block title %}  
 Автодеталь  
 {% endblock %}  
  
</head>  
{% block body %}  
  
  
 <div class="ttt" >  
  
  
 <a href="vaz"> <img src="https://static.lada.ru/images\_new/cars/about/xray/hatchback/v1\_x.jpg" alt="Ваз" width="800" >  
 </a>  
 <a href="gaz"> <img src="https://moving-expert.kyiv.ua/wp-content/webp-express/webp-images/uploads/2020/08/gazelle.jpg.webp" alt="Газ" width="800" >  
 </a>  
 <a href="gaz"> <img src="https://cdn24.img.ria.ru/images/152903/07/1529030776\_0:156:3000:1844\_1920x0\_80\_0\_0\_79bd0671b23eafa813c83bc720f2c3e9.jpg" alt="Уаз" width="800">  
 </a>  
 <a href="paz"> <img src="https://static.riafan.ru/uploads/2021/04/13/orig-paz-dom-na-kolesah-1-1618298773.jpg" alt="Паз" width="800">  
 </a>  
  
  
  
 </div>  
{% endblock %}

login.html

{% extends "base.html" %}  
  
{% block body %}  
 <h1>Авторизация</h1>  
 <form action="" method="post">  
 {{ form.hidden\_tag() }}  
 <p>  
 {{ form.email.label }}<br>  
 {{ form.email(class="form-control", type="email") }}<br>  
 {% for error in form.email.errors %}  
 <div class="alert alert-danger" role="alert">  
 {{ error }}  
 </div>  
 {% endfor %}  
 </p>  
 <p>  
 {{ form.password.label }}<br>  
 {{ form.password(class="form-control", type="password") }}<br>  
 {% for error in form.password.errors %}  
 <div class="alert alert-danger" role="alert">  
 {{ error }}  
 </div>  
 {% endfor %}  
 </p>  
 <p>{{ form.remember\_me() }} {{ form.remember\_me.label }}</p>  
 <p>{{ form.submit(type="submit", class="btn btn-primary") }}</p>  
 <div>{{ message }}</div>  
 </form>  
{% endblock %}

register.html

{% extends "base.html" %}  
{% block body %}  
<h1>Регистрация</h1>  
<form action="" method="post">  
 {{ form.hidden\_tag() }}  
 {{ form.csrf\_token }}  
 <p>  
 {{ form.email.label }}<br>  
 {{ form.email(class="form-control", type="email") }}<br>  
 {% for error in form.email.errors %}  
 <p class="alert alert-danger" role="alert">  
 {{ error }}  
 </p>  
 {% endfor %}  
 </p>  
 <p>  
 <p>  
 {{ form.password.label }}<br>  
 {{ form.password(class="form-control", type="password") }}<br>  
 {% for error in form.password.errors %}  
 <p class="alert alert-danger" role="alert">  
 {{ error }}  
 </p>  
 {% endfor %}  
 </p>  
 <p>  
 {{ form.password\_again.label }}<br>  
 {{ form.password\_again(class="form-control", type="password") }}<br>  
 {% for error in form.password\_again.errors %}  
 <p class="alert alert-danger" role="alert">  
 {{ error }}  
 </p>  
 {% endfor %}  
 </p>  
 <p> {{ form.name.label }}<br>  
 {{ form.name(class="form-control") }}<br>  
 {% for error in form.name.errors %}  
 <p class="alert alert-danger" role="alert">  
 {{ error }} </p>  
 {% endfor %}  
 </p> <p>  
 {{ form.about.label }}<br>  
 {{ form.about(class="form-control") }}<br>  
 {% for error in form.about.errors %}  
 <p class="alert alert-danger" role="alert">  
 {{ error }} </p>  
 {% endfor %}  
 </p>  
 <p>{{ form.submit(type="submit", class="btn btn-primary") }}</p>  
 {{message}}  
</form>  
{% endblock %}