Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Базовые компоненты интернет-технологий»

Отчет по рубежному контролю №2

Вариант №7 (А)

Выполнил:
студент группы ИУ5-51Б
Вардумян А.Т.

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю. Е.

Москва, 2021 г.

Задание:

Рубежный контроль представляет собой разработку веб-приложения с использованием фреймворка Django. Веб-приложение должно выполнять следующие функции:

1. Создайте проект Python Django с использованием стандартных средств Django.
2. Создайте модель Django ORM, содержащую две сущности, связанные отношением один-ко-многим в соответствии с Вашим вариантом из условий рубежного контроля №1.
3. С использованием стандартного механизма Django сгенерируйте по модели макет веб-приложения, позволяющий добавлять, редактировать и удалять данные.
4. Создайте представление и шаблон, формирующий отчет, который содержит соединение данных из двух таблиц.

Текст программы:

“bkit_rk/urls.py”:

```
from django.contrib import admin
from django.urls import path, include
from app import views

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', views.get_computers, name="main"),
    path('create/', views.create_computer, name="add"),
    path('update/<int:computer_id>', views.update_computer, name="edit"),
    path('delete/<int:computer_id>/', views.delete_computer, name="delete"),
]
```

“app/models.py”:

```
from django.db import models


class Processor(models.Model):
    name = models.CharField(max_length=256, verbose_name="processor name")
    frequency = models.FloatField(verbose_name="processor frequency")

    def __str__(self):
        return self.name


class Computer(models.Model):
    name = models.CharField(max_length=256, verbose_name="computer name")
    processor = models.ForeignKey(
        Processor,
        on_delete=models.SET_DEFAULT,
        null=True,
```

```
        default=None,
        related_name="computers"
    )

    def __str__(self):
        return self.name
```

“app/views.py”:

```
from django.shortcuts import render, redirect, get_object_or_404
from app.models import Computer, Processor
from app.forms import AddComputerForm


def get_computers(request):
    return render(request, 'report.html', {
        'data': {
            'computers': Computer.objects.all(),
            'processors': Processor.objects.all(),
        }
    })


def create_computer(request):
    if request.method == 'POST':
        form = AddComputerForm(request.POST)
        if form.is_valid():
            name = form.cleaned_data['name']
            processor_id = form.cleaned_data['processor']
            processor = Processor.objects.filter(id=processor_id)[0]
            Computer.objects.create(name=name, processor=processor)
    return redirect('main')


def update_computer(request, computer_id):
    if request.method == 'POST':
        form = AddComputerForm(request.POST)
        if form.is_valid():
            name = form.cleaned_data['name']
            processor_id = form.cleaned_data['processor']
            processor = Processor.objects.filter(id=processor_id)[0]
            Computer.objects.filter(id=computer_id).update(name=name,
processor=processor)
    return redirect('main')


def delete_computer(request, computer_id):
    computer = get_object_or_404(Computer, id=computer_id)
    computer.delete()
    return redirect('main')
```

“app/forms.py”:

```
from django import forms
```

```
class AddComputerForm(forms.Form):
    name = forms.CharField(max_length=256)
    processor = forms.IntegerField()
```

“templates/base.html”:

```
<!doctype html>
<html lang="en" class="h-100">

<head>
    <meta charset="utf-8">
    {% load static %}
    <link rel="stylesheet" type="text/css" href="{% static 'css/index.css'
%}">
    <link rel="stylesheet" type="text/css" href="{% static 'css/form.css'
%}">
    <link rel="stylesheet" type="text/css" href="{% static 'css/modal.css'
%}">
    <title>{% block title %}{% endblock %}</title>
</head>

<body>
{% block content %}{% endblock %}
<footer>Made by Vardumyan Arsen IU5-51B</footer>
</body>
</html>
```

“templates/report.html”:

```
{% extends 'base.html' %}

{% block title %}Computers{% endblock %}

{% block content %}
    <div class="main-menu">
        <ul>
            {% for computer in data.computers %}
                <li>{% include 'card.html' %}</li>
            {% empty %}
                <li>Список пуст</li>
            {% endfor %}
        </ul>
        {% include 'modal.html' %}
    </div>
{% endblock %}
```

“templates/modal.html”:

```
<button id="myBtn">Add New Computer</button>

<div id="myModal" class="modal">
    <div class="modal-content">
        <span class="close">&times;</span>
        {% include 'form.html' with url_val="add" button_name="Add" %}
    </div>
</div>
```

```
<script>
    let modal = document.getElementById("myModal");
    let btn = document.getElementById("myBtn");
    let span = document.getElementsByClassName("close")[0];

    btn.onclick = function() {
        modal.style.display = "block";
    }

    span.onclick = function() {
        modal.style.display = "none";
    }

    window.onclick = function(event) {
        if (event.target == modal) {
            modal.style.display = "none";
        }
    }
</script>
```

“templates/form.html”:

```
{% if url_param %}
    <form action="{% url url_val url_param%}" method="post" class='right-
container'>
{% else %}
    <form action="{% url url_val %}" method="post" class='right-container'>
{% endif %}
    {% csrf_token %}
    <header>
        <div class='set'>
            <div id='pets-name'>
                <label for='pets-name'>Name</label>
                <input placeholder="Computer's name" name="name" type='text'>
            </div>
        </div>
        <div class='pets-weight'>
            <label>Processor</label>
            <div class='radio-container'>
                {% for processor in data.processors%}
                    <input checked='' id="{{ url_param }}{{ processor.id }}"
name="processor" type='radio' value={{ processor.id }}>
                    <label for="{{ url_param }}{{ processor.id }}">{{
processor.name }} ({{ processor.frequency }})</label>
                {% endfor %}
            </div>
        </div>
        <br>
        <button type="submit">{{ button_name }}</button>
    </header>
</form>
```

“templates/card.html”:

```
<div class="container">
    <div class="card">
        <div class="card__body">
            <h4>{{ computer.name }}</h4>
            <p>{{ computer.processor }}</p>
            <p>{{ computer.processor.frequency }}</p>
        </div>
        <div class="card__footer">
            {% include 'form.html' with url_val='edit' url_param=computer.id
button_name="Edit" %}
            {% include 'button.html' with url_val="delete"
url_param=computer.id title="Delete" %}
        </div>
    </div>
</div>
```

“templates/button.html”:

```
{% if url_param %}
    <a href="{% url url_val url_param %}" class="link">
{% else %}
    <a href="{% url url_val %}" class="link">
{% endif %}
    <span class="mask">
        <div class="link-container">
            <span class="link-title1 title">{{ title }}</span>
            <span class="link-title2 title">{{ title }}</span>
        </div>
    </span>
</a>
```

“bkit_rk/setting.py” (фрагмент настройки БД):

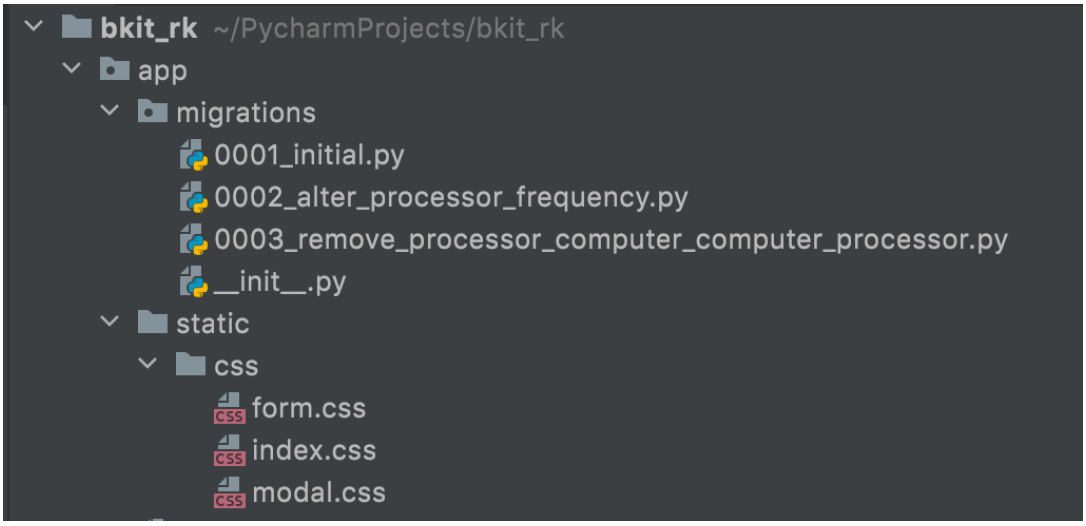
```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'rk_db',
        'USER': 'arsenvardumyan',
        'PASSWORD': '',
        'HOST': 'localhost',
        'PORT': 5432,
    }
}
```

“fill_db.py” (скрипт для заполнения БД):

```
import psycopg2

db = psycopg2.connect(
    host="localhost",
    user="arsenvardumyan",
    dbname="rk_db"
)

c = db.cursor()
```

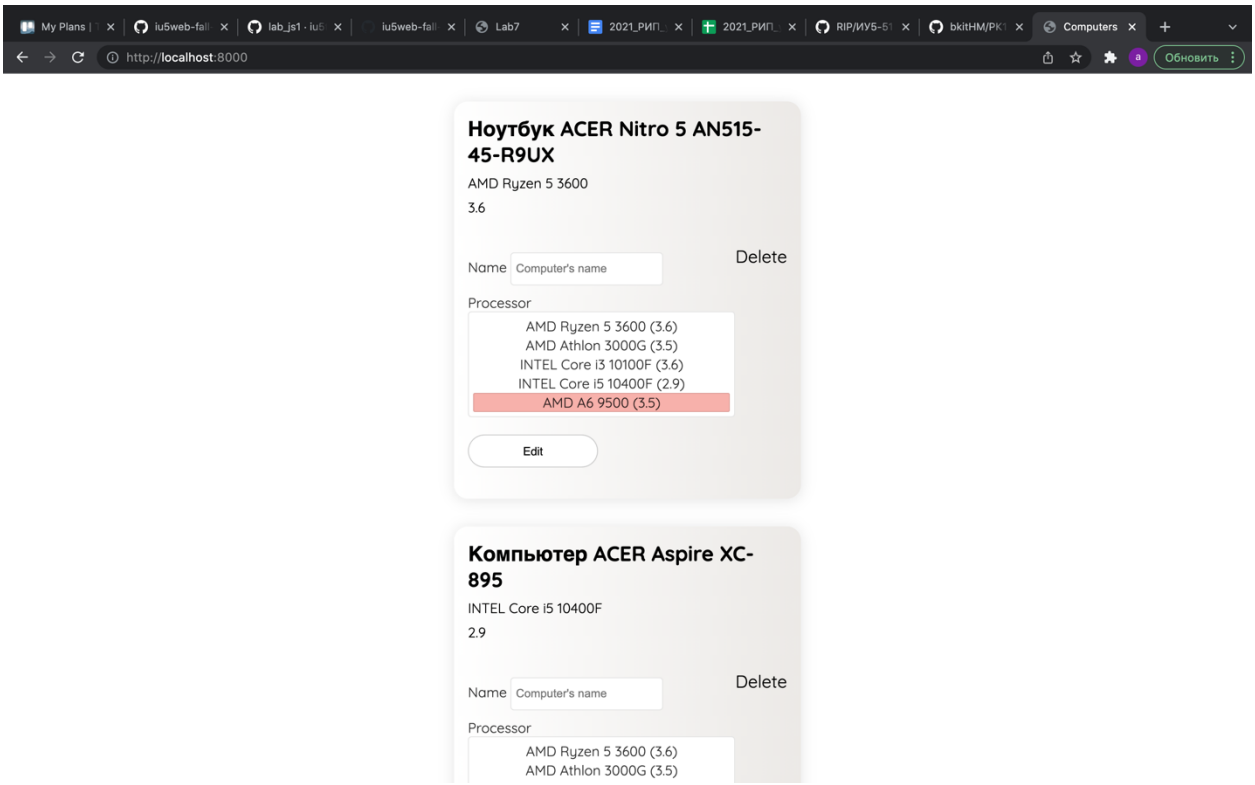
```
# c.execute('INSERT INTO "app_processor" (name, frequency) VALUES (%s, %s);',
#           ('AMD Ryzen 5 3600', 3.6))
#
# c.execute('INSERT INTO "app_processor" (name, frequency) VALUES (%s, %s);',
#           ('AMD Athlon 3000G', 3.5))
#
# c.execute('INSERT INTO "app_processor" (name, frequency) VALUES (%s, %s);',
#           ('INTEL Core i3 10100F', 3.6))
#
# c.execute('INSERT INTO "app_processor" (name, frequency) VALUES (%s, %s);',
#           ('INTEL Core i5 10400F', 2.9))
#
# c.execute('INSERT INTO "app_processor" (name, frequency) VALUES (%s, %s);',
#           ('AMD A6 9500', 3.5))
#
# db.commit()

c.execute('INSERT INTO "app_computer" (name, processor_id) VALUES (%s, %s);',
          ('Ноутбук ACER Nitro 5 AN515-45-R9UX', 1))

c.execute('INSERT INTO "app_computer" (name, processor_id) VALUES (%s, %s);',
          ('Компьютер ACER Aspire XC-895', 4))

c.execute('INSERT INTO "app_computer" (name, processor_id) VALUES (%s, %s);',
          ('Компьютер ACER Aspire XC-830', 2))

c.execute('INSERT INTO "app_computer" (name, processor_id) VALUES (%s, %s);',
          ('Ноутбук LENOVO IdeaPad S145-15API', 1))

c.execute('INSERT INTO "app_computer" (name, processor_id) VALUES (%s, %s);',
          ('Компьютер IRU Home 615', 3))

c.execute('INSERT INTO "app_computer" (name, processor_id) VALUES (%s, %s);',
          ('Ноутбук HP 15-dw1126ur', 5))

db.commit()

c.close()
db.close()
```

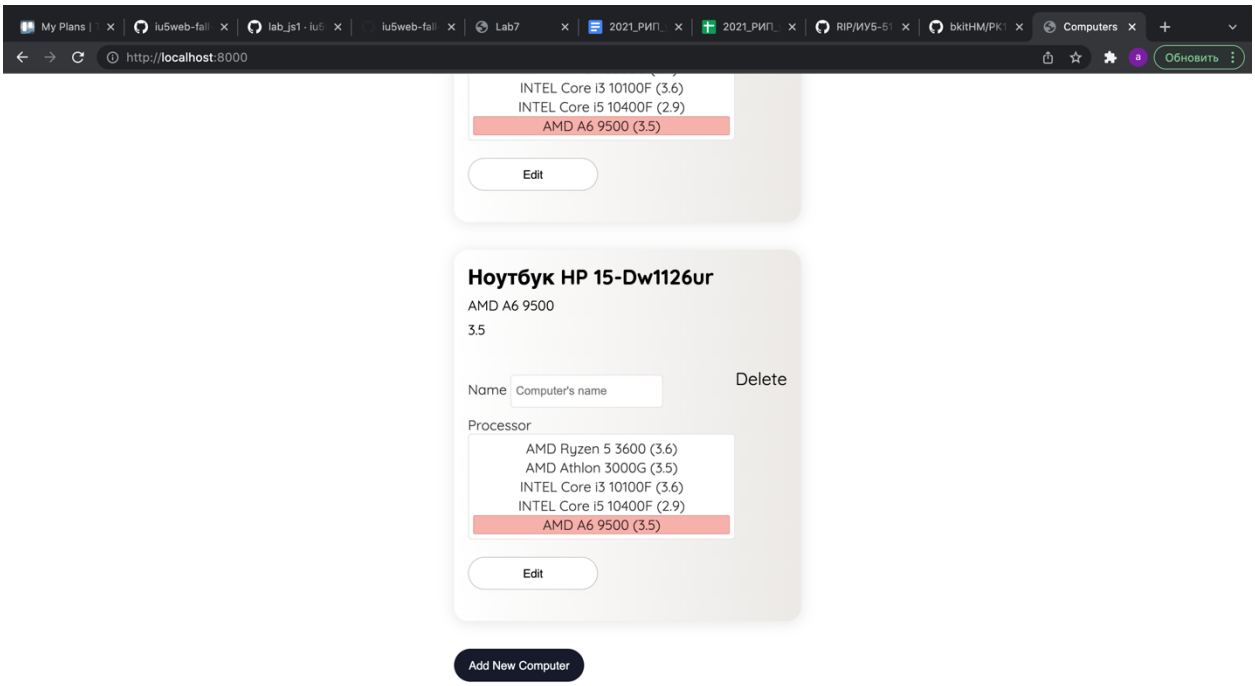
Статика и миграции:

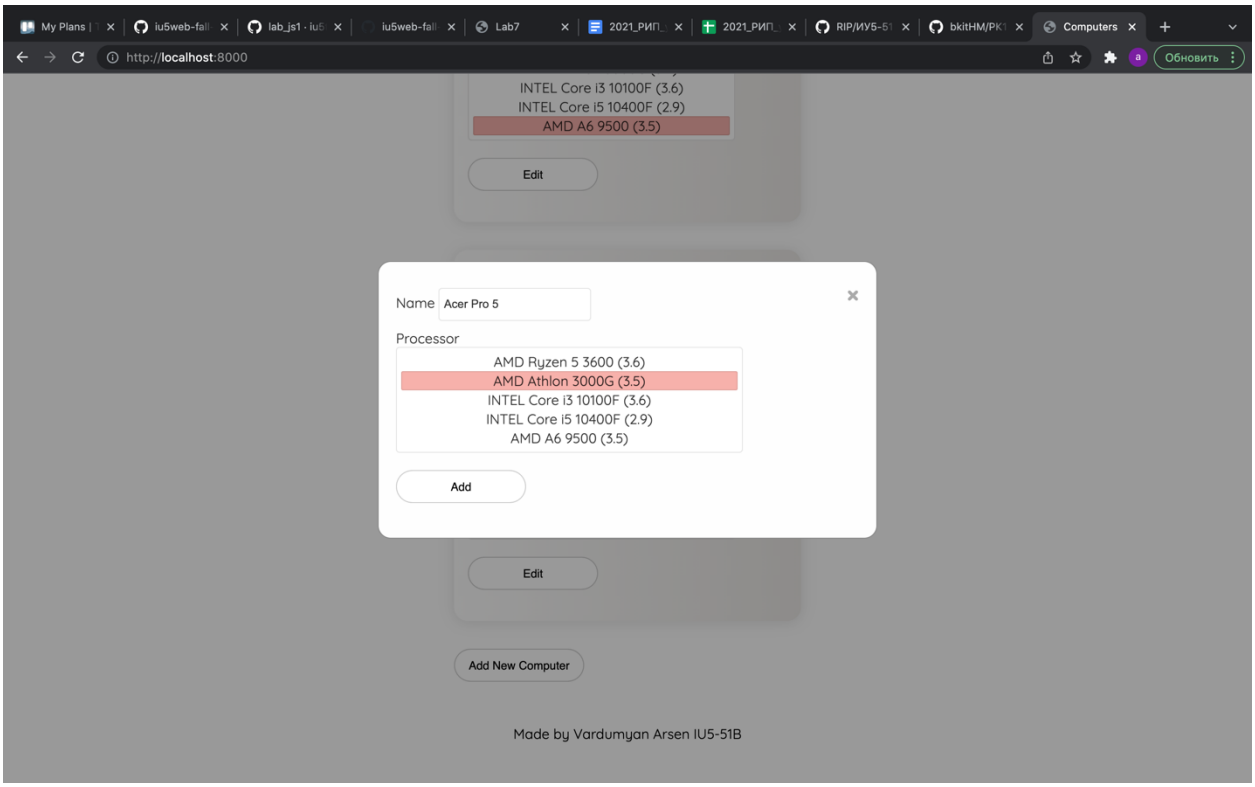


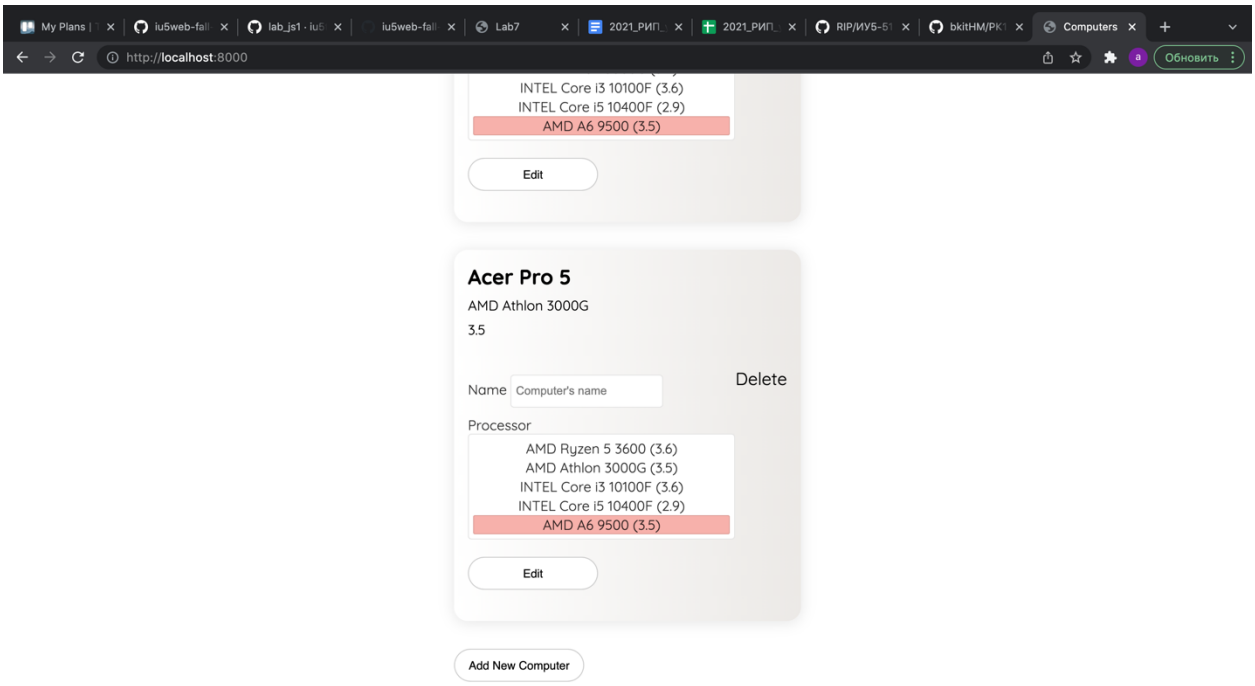
ЕМ-схема БД:

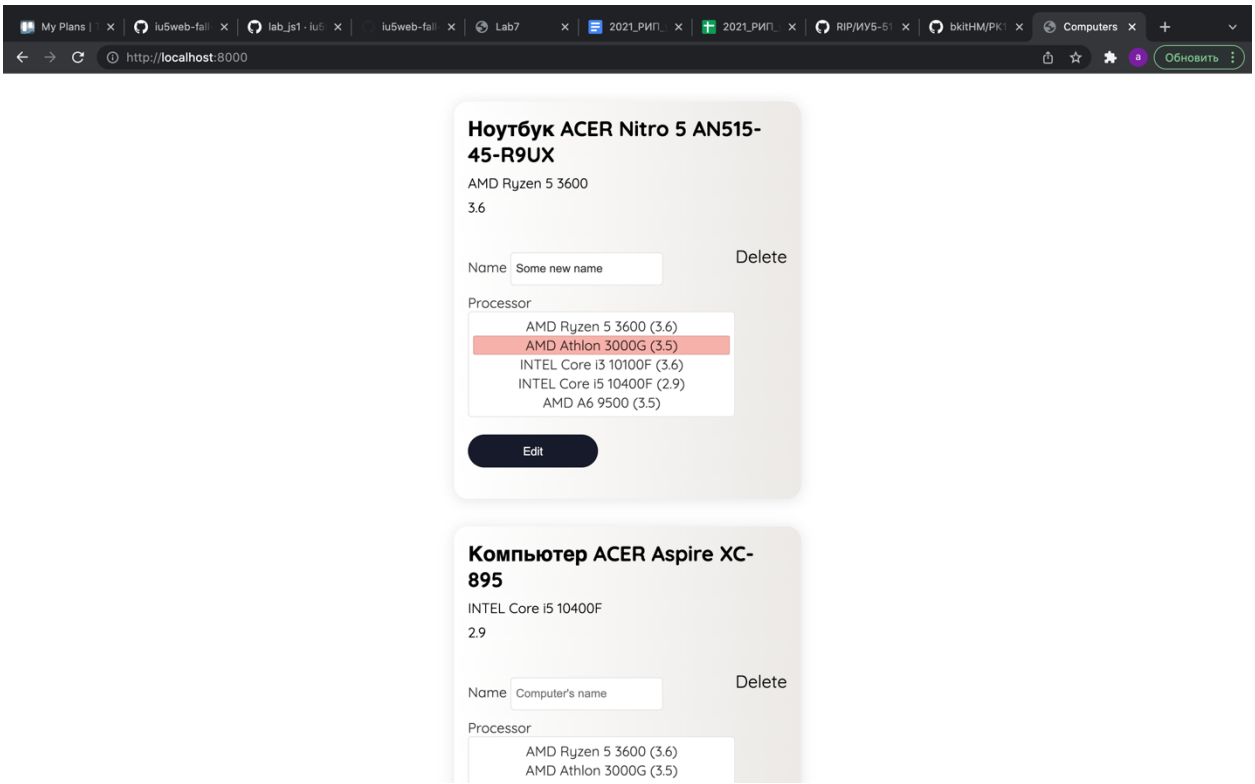
Результаты программы:




My Plans
iu5web-fall
lab_js1 - iu5
iu5web-fall
Lab7
2021_РИП
2021_РИП
RIP/ИУ5-51
bkitHM/PK1
Computers
http://localhost:8000
Обновить
INTEL Core i3 10100F (3.6)
INTEL Core i5 10400F (2.9)
AMD A6 9500 (3.5)
Edit
Ноутбук HP 15-Dw1126ur
AMD A6 9500
3.5
Delete
Name
Computer's name
Processor
AMD Ryzen 5 3600 (3.6)
AMD Athlon 3000G (3.5)
INTEL Core i3 10100F (3.6)
INTEL Core i5 10400F (2.9)
AMD A6 9500 (3.5)
Edit
Add New Computer
Made by Vardumyan Arsen IU5-51B


My Plans
iu5web-fall
lab_js1 - iu5
iu5web-fall
Lab7
2021_РИП
2021_РИП
RIP/ИУ5-51
bkitHM/PK1
Computers
http://localhost:8000
Обновить
INTEL Core i3 10100F (3.6)
INTEL Core i5 10400F (2.9)
AMD A6 9500 (3.5)
Edit
Name
Acer Pro 5
Processor
AMD Ryzen 5 3600 (3.6)
AMD Athlon 3000G (3.5)
INTEL Core i3 10100F (3.6)
INTEL Core i5 10400F (2.9)
AMD A6 9500 (3.5)
Add
Edit
Add New Computer
Made by Vardumyan Arsen IU5-51B


My Plans
iu5web-fall
lab_js1 - iu5
iu5web-fall
Lab7
2021_РИП
2021_РИП
RIP/ИУ5-51
bkitHM/PK1
Computers
http://localhost:8000
Обновить
INTEL Core i3 10100F (3.6)
INTEL Core i5 10400F (2.9)
AMD A6 9500 (3.5)
Edit
Acer Pro 5
AMD Athlon 3000G
3.5
Delete
Name
Computer's name
Processor
AMD Ryzen 5 3600 (3.6)
AMD Athlon 3000G (3.5)
INTEL Core i3 10100F (3.6)
INTEL Core i5 10400F (2.9)
AMD A6 9500 (3.5)
Edit
Add New Computer
Made by Vardumyan Arsen IU5-51B


My Plans
iu5web-fall
lab_js1 - iu5
iu5web-fall
Lab7
2021_РИП
2021_РИП
RIP/ИУ5-51
bkitHM/PK1
Computers
http://localhost:8000
Обновить
Ноутбук ACER Nitro 5 AN515-45-R9UX
AMD Ryzen 5 3600
3.6
Delete
Name
Some new name
Processor
AMD Ryzen 5 3600 (3.6)
AMD Athlon 3000G (3.5)
INTEL Core i3 10100F (3.6)
INTEL Core i5 10400F (2.9)
AMD A6 9500 (3.5)
Edit
Компьютер ACER Aspire XC-895
INTEL Core i5 10400F
2.9
Delete
Name
Computer's name
Processor
AMD Ryzen 5 3600 (3.6)
AMD Athlon 3000G (3.5)