

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ  
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»  
(БГТУ им. В.Г. Шухова)

Кафедра программного обеспечения вычислительной техники и автоматизированных  
систем

Индивидуальное домашнее задание  
по дисциплине: Технологии web-программирования  
тема: «**Разработка Web-приложения**»

Выполнила: ст. группы ПВ-42  
Петрова Алина Ильинична  
Проверил: Картамышев С.В.

Белгород 2021 г.

## Оглавление

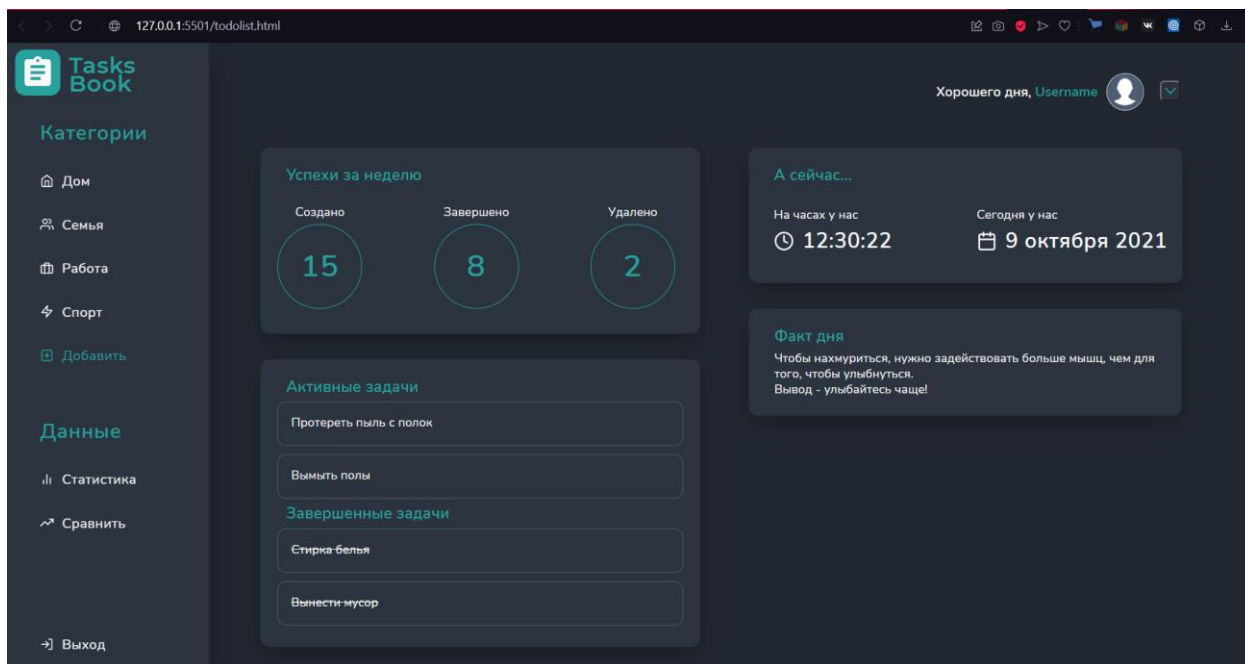
1. HTML. Разработка макетов и верстка шаблонов web-приложения с помощью языков HTML и CSS.....	3
2. Клиентское программирование. React js.....	21
3. Серверное программирование.....	25
4. Разработка и проектирование базы данных .....	29
5. REST API.....	31

## 1. HTML. Разработка макетов и верстка шаблонов web-приложения с помощью языков HTML и CSS

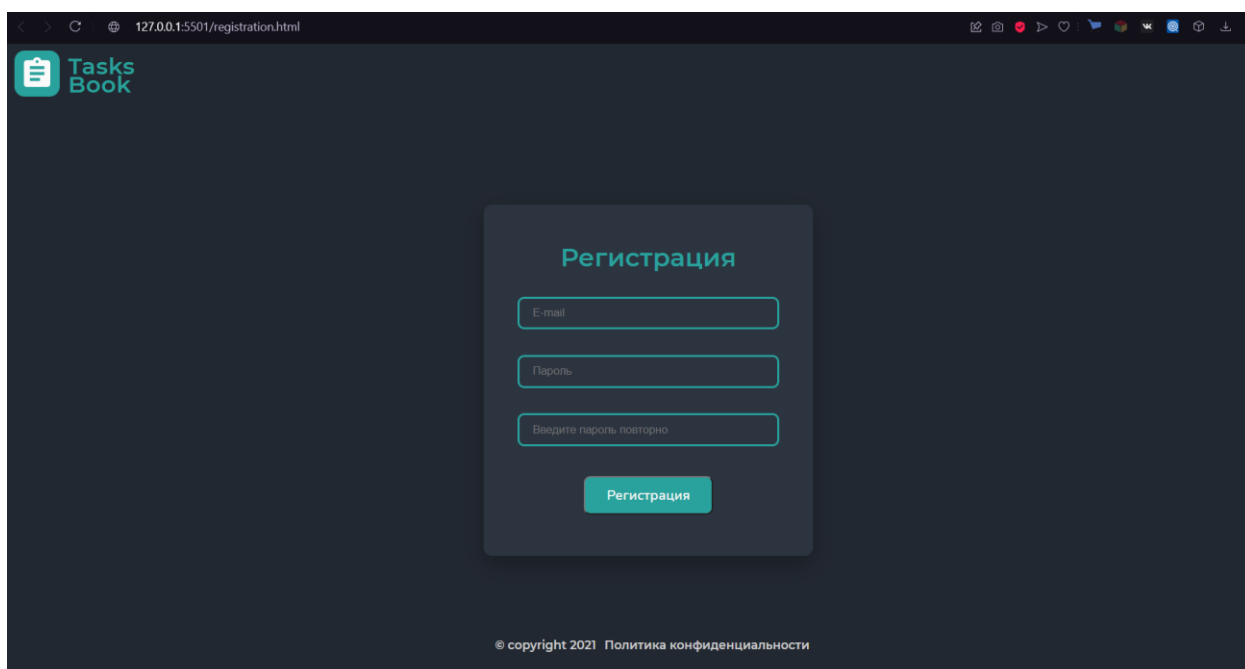
В ходе подготовки к работе мною был выбран предмет разработки, а именно web-приложение «TO-DO List».

В ходе выполнения первой лабораторной работы были разработаны следующие страницы:

Главная страница



Страница регистрации



## Страница входа

The screenshot shows a web browser window with the address bar displaying '127.0.0.1:5501/login.html'. The page features a dark blue header with the 'Tasks Book' logo on the left. In the center, there is a light blue box titled 'Вход в аккаунт' (Login to account). Inside this box, there are two input fields: 'E-mail' and 'Пароль' (Password). Below these fields is a blue button labeled 'Войти' (Login). At the bottom of the box, there is a link that says 'Ещё нет аккаунта? Регистрация' (Don't have an account? Registration). At the very bottom of the page, there is a small copyright notice: '© copyright 2021 Политика конфиденциальности'.

## Страница редактирования аккаунта

The screenshot shows a web browser window with the address bar displaying '127.0.0.1:5501/account.html'. The page has a dark blue header with the 'Tasks Book' logo on the left. On the right side of the header, there is a greeting 'Хорошего дня, Username' followed by a user profile icon and a checkmark. A left sidebar contains a list of categories: 'Дом', 'Семья', 'Работа', 'Спорт', 'Добавить', and 'Данные'. Below these are links for 'Статистика' and 'Сравнить', and a 'Выход' (Logout) button at the bottom. The main content area is divided into three sections. The first section on the left shows a user profile icon with a 'Изменить фото' (Change photo) link below it. The second section in the center is titled 'Ваш никнейм' (Your nickname) and contains three input fields: 'Никнейм', 'Изменить email', and 'Изменить пароль'. Below these fields is a blue button labeled 'Сохранить изменения' (Save changes). The third section on the right is titled 'А сейчас...' (Now...) and displays the current time '12:30:22' and the date '9 октября 2021'. Below this is a section titled 'Факт дня' (Fact of the day) with a motivational quote: 'Чтобы нахмуриться, нужно задействовать больше мышц, чем для того, чтобы улыбнуться. Вывод - улыбайтесь чаще!' (To frown, you need to use more muscles than to smile. Conclusion - smile more!).

## Пример исходного кода

Файл todolist.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">

  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link
href="https://fonts.googleapis.com/css2?family=Montserrat:wght@600&family=Nunito:
wght@300;400;600&display=swap" rel="stylesheet">

  <link rel="stylesheet" href="css/style.css">
</head>

<body>

  <section class="Sidebar">
    <header>
      <section class="logoContainer">
        <section class="logo">
          
        </section>
        <section class="logoText">
          <h2>Tasks<br/>Book</h2>
        </section>
      </section>
    </header>

    <form class="Category">
      <ul>
        <lh>Категории</lh>
        <li class="home"><a
href="#">Дом</a></li>
        <li class="family"><a
href="#">Семья</a></li>
        <li class="work"><a
href="#">Работа</a></li>
        <li class="sport"><a
href="#">Спорт</a></li>
        <li class="add"><a
href="#">Добавить</a></li>
      </ul>

      <ul>
        <lh>Данные</lh>
        <li class="stat"><a
href="#">Статистика</a></li>
```

```

        <li class="trend"><a href="#">Сравнить</a></li>
    </ul>

    <ul>
        <li class="out"><a
href="#">Выход</a></li>
    </ul>
</form>
</section>

<main class="Main">

    <section class="Left">
        <div class="Success">
            <h2>Успехи за неделю</h2>

            <div class="Circles">
                <div class="ContCirc">
                    <h3>Создано</h3>
                    <div class="Circle">
                        <h4>15</h4>
                    </div>
                </div>
                <div class="ContCirc">
                    <h3>Завершено</h3>
                    <div class="Circle">
                        <h4>8</h4>
                    </div>
                </div>
                <div class="ContCirc">
                    <h3>Удалено</h3>
                    <div class="Circle">
                        <h4>2</h4>
                    </div>
                </div>
            </div>
        </div>
    </div>

    <div class="Tasks">

        <h2>Активные задачи</h2>
        <div class="Task">
            <span class="active">Протереть пыль с полок</span>
        </div>
        <div class="Task">
            <span class="active">Вымыть полы</span>
        </div>

        <h2>Завершенные задачи</h2>

```

```

        <div class="Task">
            <span class="done">Стирка белья</span>
        </div>
        <div class="Task">
            <span class="done">Вынести мусор</span>
        </div>
    </div>

</section>

<section class="Right">
    <div class="TimeCont">
        <h2>А сейчас...</h2>
        <form class="Datetime">
            <div class="Time">
                <h3>На часах у нас</h3>
                <span class="Timedate">12:30:22</span>
            </div>
            <div class="Date">
                <h3>Сегодня у нас</h3>
                <span class="Timedate">9 октября 2021</span>
            </div>
        </form>
    </div>

    <div class="Fact">
        <h2>Факт дня</h2>
        <h3>Чтобы нахмуриться, нужно задействовать больше мышц, чем
для того, чтобы улыбнуться.<br>Вывод - улыбайтесь чаще!</h3>
    </div>
</section>

</main>

    <p class="User">Хорошего дня,&nbsp;<a href="#"> Username</a></p>

    <div class="Menu">
        <button onclick="myFunction()" class="Menubutton"></button>
        <div id="myDropdown" class="dropdown-content">
            <a href="#">Ссылка 1</a>
            <a href="#">Ссылка 2</a>
            <a href="#">Ссылка 3</a>
        </div>
    </div>

</body>

```

## Файл style.css

```
.Sidebar {  
  
    position: absolute;  
    width: 240px;  
    height: 752px;  
    left: 0px;  
  
    display: flex;  
    flex-direction: column;  
    align-items: flex-start;  
  
    /* Темная тема/Карточка */  
  
    background: #2C3440;  
  
    filter: drop-shadow(0px 10px 25px rgba(0, 0, 0, 0.25));  
}  
  
.Category {  
  
    display: flex;  
    flex-direction: column;  
    align-items: flex-start;  
    padding: 0px;  
  
    position: static;  
    width: 212px;  
    height: 283px;  
    left: auto;  
    top: 0x;  
}  
.Category ul li {  
  
    flex: none;  
    order: 1;  
    flex-grow: 0;  
    margin: 30px 0px;  
  
    list-style-type: none;  
    font-family: Nunito;  
    font-style: normal;  
    font-weight: normal;  
    font-size: 16px;
```



```

    line-height: 22px;
    /* identical to box height */

    letter-spacing: 0.025em;
}

.Category ul lh {

    font-family: Nunito;
    font-style: normal;
    font-weight: 600;
    font-size: 24px;
    line-height: 33px;
    /* identical to box height */

    letter-spacing: 0.03em;

    /* Акцент */

    color: #29A19C;
}

.Category ul li a {
    color: #F9F9F9;
    text-decoration: none;
}

.Category .add a{

    flex: none;
    order: 4;
    flex-grow: 0;
    margin: 30px 0px;
    color: #29A19C;
    text-decoration: none;
}

.Category li a:hover {

    text-decoration: underline;
}

.icon {

    width: 16px;
    padding-right: 10px;
    position: relative;
    top: 2px;

```

```

}

.Category .out {

    margin-top: 60px;
}

.Main {

    position: absolute;
    width: 1069px;
    height: 715px;
    left: 301px;
    top: 94px;
}

.Success {

    display: flex;
    flex-direction: column;
    align-items: flex-start;
    padding: 20px;

    position: static;
    width: 480px;
    height: 180px;
    left: 0px;
    top: 0px;

    /* Темная тема/Карточка */

    background: #2C3440;
    /* Тень */

    box-shadow: 0px 10px 25px rgba(0, 0, 0, 0.25);
    border-radius: 10px;

    /* Inside Auto Layout */

    flex: none;
    order: 0;
    flex-grow: 0;
    margin: 30px 0px;
}

.Main h2 {

    display: flex;

```

```

flex-direction: row;
justify-content: space-between;
align-items: center;
padding: 0px;

position: static;
width: 470px;
height: 25px;

font-family: Nunito;
font-style: normal;
font-weight: 600;
font-size: 18px;
line-height: 25px;
/* identical to box height */

letter-spacing: 0.02em;

/* Акцент */

color: #29A19C;
/* Inside Auto Layout */

flex: none;
order: 0;
align-self: stretch;
flex-grow: 0;
margin: 0px 10px;
}

.Circles {

display: flex;
flex-direction: row;
justify-content: space-between;
align-items: flex-start;
padding: 0px;

position: static;
width: 470px;
height: 126px;
left: 20px;
top: 20px;

/* Inside Auto Layout */

flex: none;
order: 1;

```

```

    align-self: stretch;
    flex-grow: 0;
    margin: 0px 0px;
}

.Circle {

    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: center;
    padding: 0px;

    position: static;
    width: 100px;
    height: 100px;
    left: 0px;
    top: 0px;

    /* Темная тема/Карточка */

    background: #2C3440;
    /* Акцент */

    border: 1px solid #29A19C;
    box-sizing: border-box;
    border-radius: 200px;
}

.ContCirc {

    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: center;
    padding: 0px;

    position: static;
    width: 100px;
    height: 142px;
    left: 0px;
    top: 0px;

    /* Inside Auto Layout */

    flex: none;

```

```

    order: 0;
    flex-grow: 0;
    margin: 10px 0px;

    font-style: normal;
    font-weight: 600;
    font-size: 12px;
    line-height: 16px;
    letter-spacing: 0.02em;

    /* Темная тема/Белый */

    color: #F9F9F9;
}

.ContCirc h4 {

    font-family: Nunito;
    font-style: normal;
    font-weight: 600;
    font-size: 36px;
    line-height: 49px;
    letter-spacing: 0.02em;

    /* Акцент */

    color: #29A19C;
}

.Tasks {
    display: flex;
    flex-direction: column;
    align-items: flex-start;
    padding: 20px;

    position: static;
    width: 480px;
    height: 300px;
    left: 0px;
    top: 241px;

    /* Темная тема/Карточка */

    background: #2C3440;
    /* Тень */

    box-shadow: 0px 10px 25px rgba(0, 0, 0, 0.25);

```

```

border-radius: 10px;

/* Inside Auto Layout */

flex: none;
order: 1;
flex-grow: 0;
margin: 30px 0px;
}

.Task {

display: flex;
flex-direction: row;
align-items: flex-start;
padding: 0px;

position: static;
width: 480px;
height: 49px;
left: 0px;
top: 59px;

background: #2C3440;
border: 1px solid rgba(249, 249, 249, 0.2);
box-sizing: border-box;
border-radius: 10px;

/* Inside Auto Layout */

flex: none;
order: 0;
flex-grow: 1;
margin: 5px 0px;
}

.Task span {

position: static;
width: 460px;
height: 19px;
left: 15px;
top: 15px;

font-family: Nunito;
font-style: normal;
font-weight: normal;
font-size: 14px;

```

```

    line-height: 19px;
    /* identical to box height */

    letter-spacing: 0.02em;

    /* Темная тема/Белый */

    color: #F9F9F9;

    /* Inside Auto Layout */

    flex: none;
    order: 1;
    flex-grow: 0;
    margin: 15px;
}

.done {
    text-decoration: line-through;
}

.Right {
    display: flex;
    flex-direction: column;
    align-items: center;
    padding: 0px;

    position: absolute;
    width: 473px;
    height: 607px;
    left: 597px;
    top: 0px;
}

.TimeCont {
    display: flex;
    flex-direction: column;
    align-items: flex-start;
    padding: 20px;

    position: static;
    width: 473px;
    height: 120px;
    left: 0px;
    top: 0px;

    background: #2C3440;
    /* Тень */

```

```

    box-shadow: 0px 10px 25px rgba(0, 0, 0, 0.25);
    border-radius: 10px;

    /* Inside Auto Layout */

    flex: none;
    order: 0;
    flex-grow: 0;
    margin: 30px 0px;
}

.Datetime {
    display: flex;
    flex-direction: row;
    justify-content: space-between;
    align-items: flex-start;
    padding: 0px;

    position: static;
    width: 433px;
    height: 58px;
    left: 20px;

    /* Inside Auto Layout */

    flex: none;
    order: 1;
    align-self: stretch;
    flex-grow: 0;
    margin: 20px 0px;
}

h3 {
    font-family: Nunito;
    font-style: normal;
    font-weight: normal;
    font-size: 14px;
    line-height: 19px;
    /* identical to box height */

    letter-spacing: 0.02em;

    color: #F9F9F9;

    /* Inside Auto Layout */

    flex: none;
    order: 0;
    flex-grow: 0;

```



```

    margin: 5px 10px;
  }

.icon1 {
  position: static;
  width: 24px;
  height: 24px;
  left: 0px;
  top: 5px;

  /* Inside Auto Layout */

  flex: none;
  order: 0;
  flex-grow: 0;
  margin: 0px 10px;

}

.Timedate {
  display: flex;
  flex-direction: row;
  align-items: center;
  padding: 0px;

  position: static;
  width: 240px;
  height: 34px;
  left: 0px;
  top: 0px;

  font-family: Nunito;
  font-style: normal;
  font-weight: 600;
  font-size: 25px;
  line-height: 34px;
  /* identical to box height */

  letter-spacing: 0.02em;

  color: #F9F9F9;

  /* Inside Auto Layout */

  flex: none;
  order: 1;
  flex-grow: 0;
  margin: 5px 0px;

```

```

}

.Fact {
    display: flex;
    flex-direction: column;
    align-items: flex-start;
    padding: 20px;

    position: static;
    width: 472px;
    height: 87px;
    left: 0.5px;
    top: 73px;

    background: #2C3440;
    /* Тень */

    box-shadow: 0px 10px 25px rgba(0, 0, 0, 0.25);
    border-radius: 10px;

    /* Inside Auto Layout */

    flex: none;
    order: 1;
    flex-grow: 0;
    margin: 0px 0px;
}

.User {
    display: flex;
    flex-direction: row;
    align-items: center;
    padding: 0px;

    position: absolute;
    width: 294px;
    height: 44px;
    left: 1100px;
    top: 20px;

    font-family: Nunito;
    font-style: normal;
    font-weight: 600;
    font-size: 16px;
    line-height: 22px;
    /* identical to box height */

    /* Темная тема/Белый */

```

```

    color: #F9F9F9;

    /* Inside Auto Layout */

    flex: none;
    order: 0;
    flex-grow: 0;

}

.User a {
    color: #29A19C;
    text-decoration: none;
}

.Usericon {

    position: relative;
    width: 42px;
    height: 42px;
    top: 0px;

    background: url(.png);
    border-radius: 50%;
    border: 1px solid #F0F0F0;

    /* Inside Auto Layout */

    flex: none;
    order: 1;
    flex-grow: 0;
    margin: 10px;

}

.show {display: block;}

.Menubutton {
    position: relative;
    width: 24px;
    height: 24px;
    left: 1345px;
    top: 10px;
    padding: 3px;

    background: #2C3440;
    /* Smooth shadow */

```

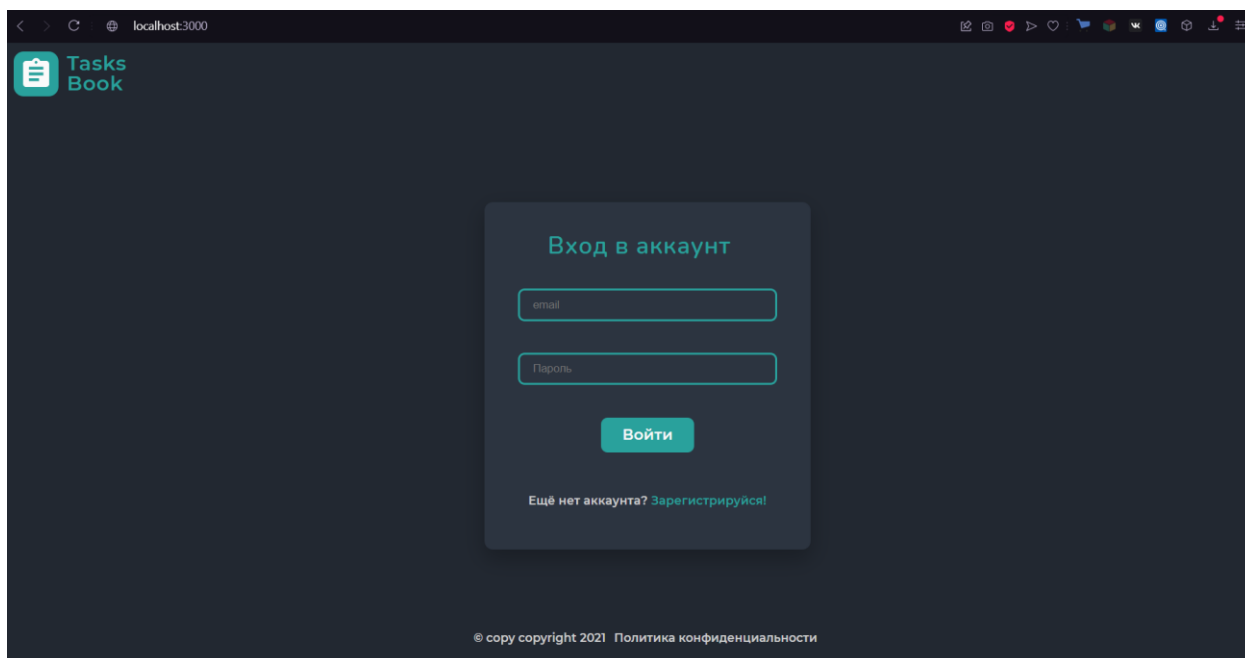
```
    box-shadow: 0px 10px 25px rgba(29, 52, 54, 0.08);
    border-radius: 5px;

    flex: none;
    order: 0;
    flex-grow: 0;
    margin: 35px 20px;
}
```

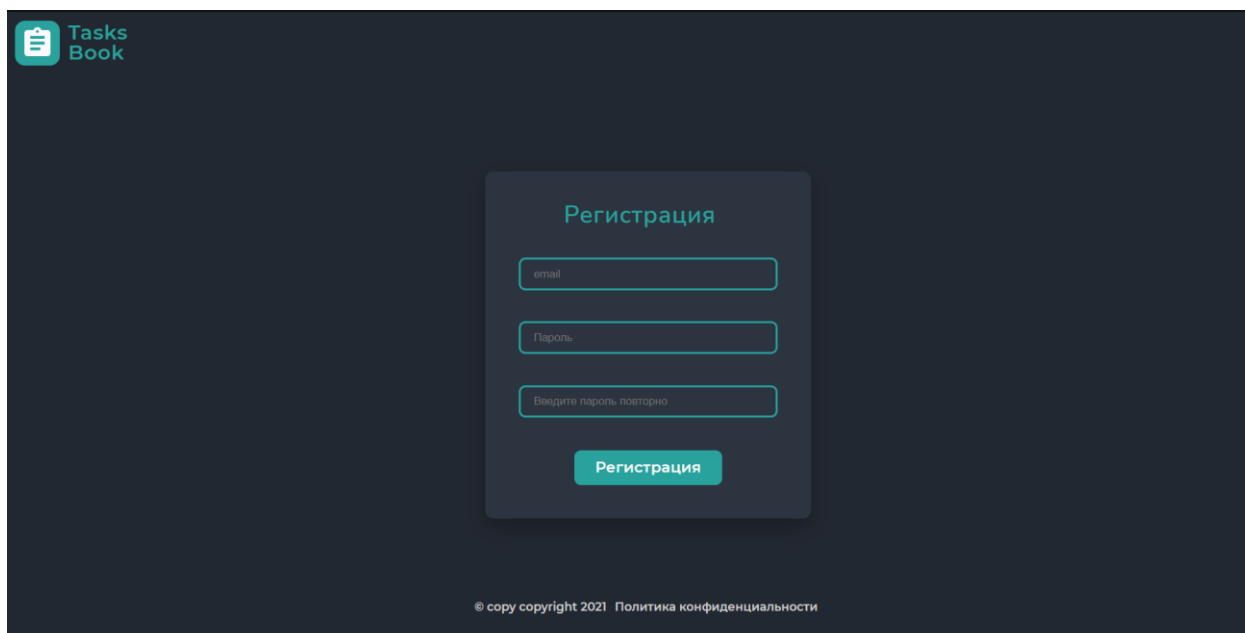
## 2. Клиентское программирование. React js

В ходе выполнения работы были изучены основы разработки frontend-приложений на языке JavaScript, развернуто базовое приложение на фреймворке React.js, в него перенесены страницы из 1ой лабораторной работы, разделены на компоненты.

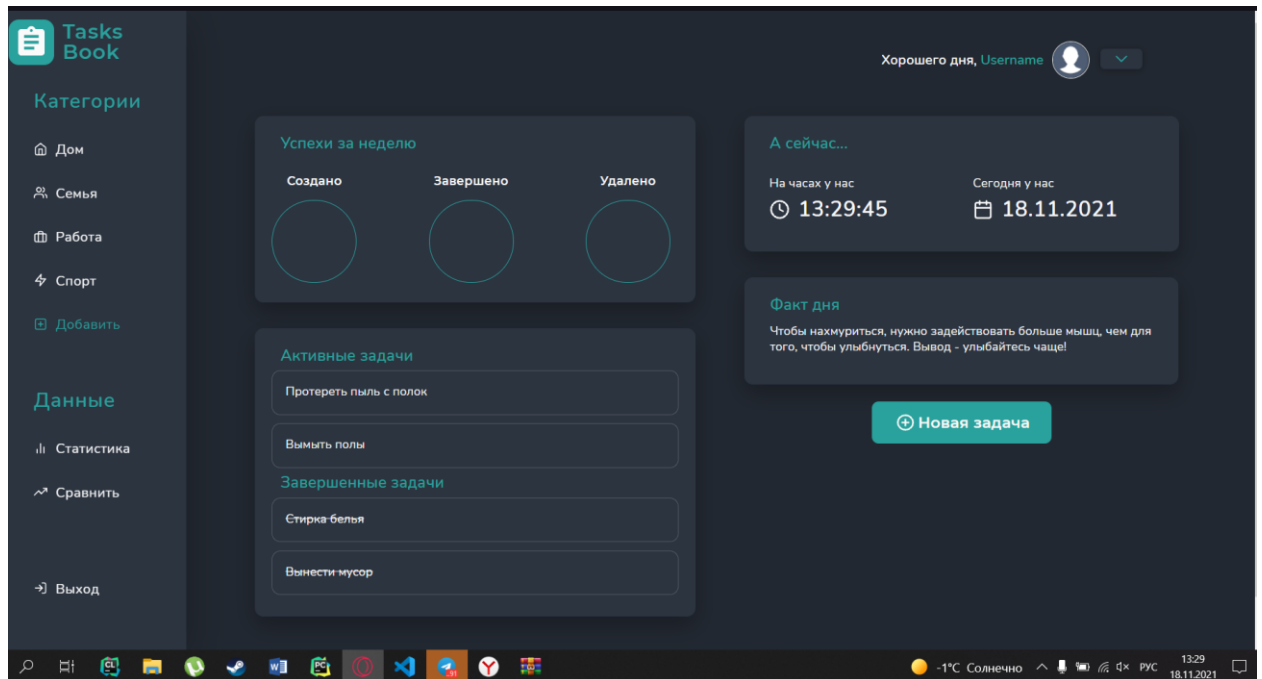
Первая страница, которую видит пользователь – страница входа.



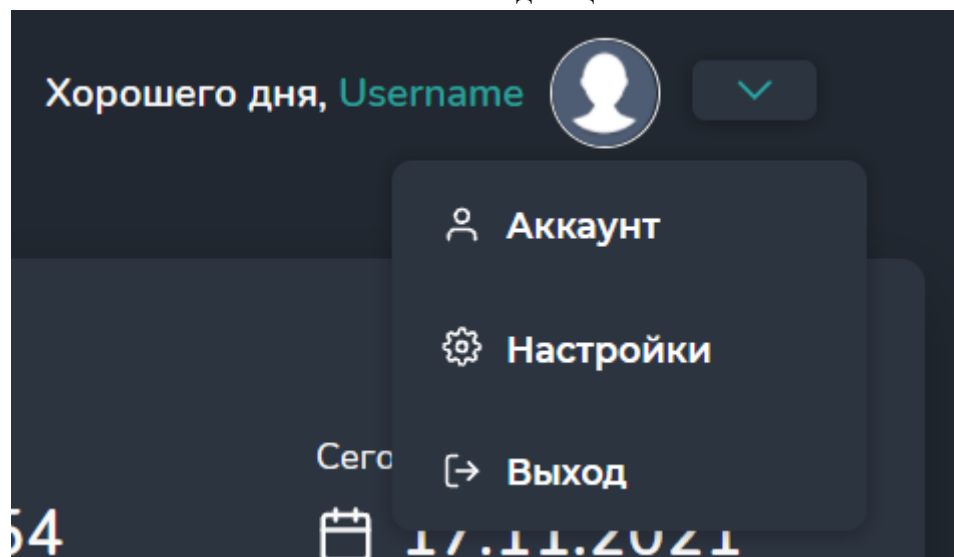
Если у пользователя еще нет аккаунта, он может перейти на страницу с регистрацией



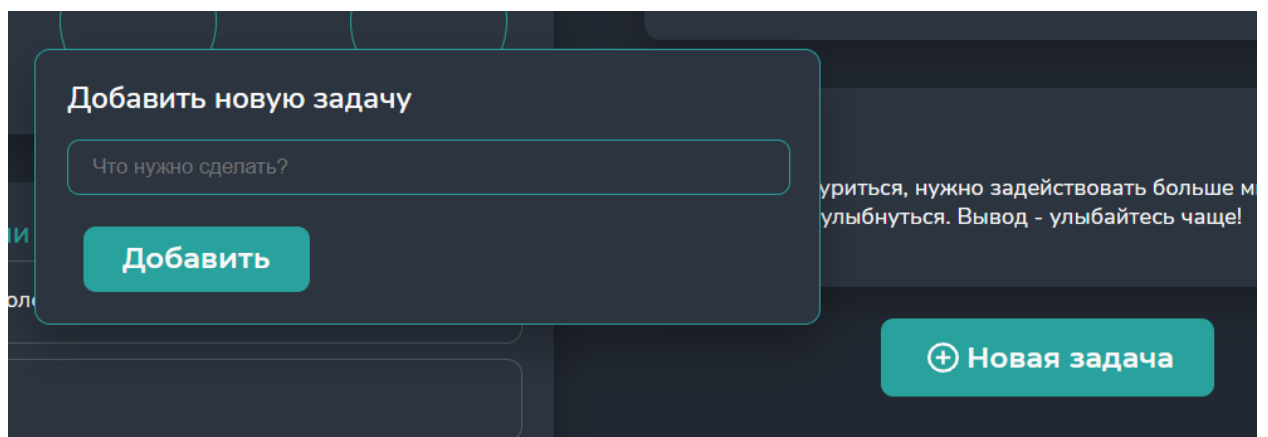
## Основная страница со списком делом и дополнительной информацией



## Реализовано выпадающее меню



## Реализовано модальное окно с добавлением задачи



## Страница для редактирования данных пользователя

Tasks Book

Категории

Дом

Семья

Работа

Спорт

Добавить

Данные

Статистика

Сравнить

Выход

Хорошего дня, Username

Ваш никнейм

Изменить фото

Изменить email

Изменить пароль

Никнейм

E-mail

Пароль

Сохранить изменения

А сейчас...

На часах у нас  
21:21:27

Сегодня у нас  
17.11.2021

Факт дня

Чтобы нахмуриться, нужно задействовать больше мышц, чем для того, чтобы улыбнуться. Вывод - улыбайтесь чаще!

## Пример исходного кода

### registration.js

```
import React from 'react';
import './style.css';
import Logo from '../UI/logo';
import Navigation from '../UI/navigation/navigation';
import Cardreg from '../UI/cardreg';

export default function registration() {
  return (
    <div>
      <Logo />
      <Cardreg name="Регистрация" />
      <Navigation />
    </div>
  )
}
```

### cardreg.js

```
import React from 'react'
import Button from '../button/button';
import Input from '../input'
import './style.css';

export default function Cardreg(props) {
  return (
    <section className="reg-container">
      <h2> {props.name} </h2>
      <Input name="email"/>
      <Input name="Пароль"/>
      <Input name="Введите пароль повторно"/>
      <Button name="Регистрация" />
    </section>
  )
}
```

### cardreg.js

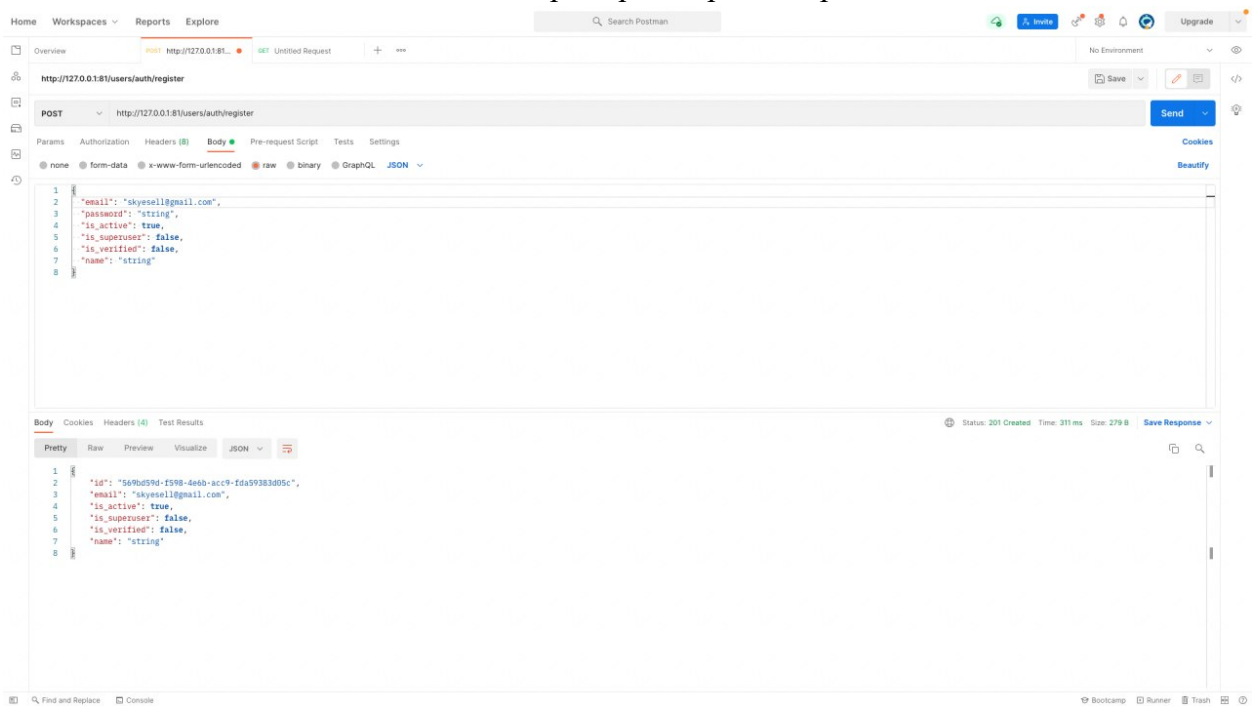
```
import React from 'react';
import './style.css';
import logo from './logo.svg';

export default function Logo() {
  return (
    <section className="logoContainer">
      <section className="logo">
        <img src={logo} alt="logo"/>
      </section>
      <section className="logoText">
        <h2>Tasks<br/>Book</h2>
      </section>
    </section>
  )
}
```

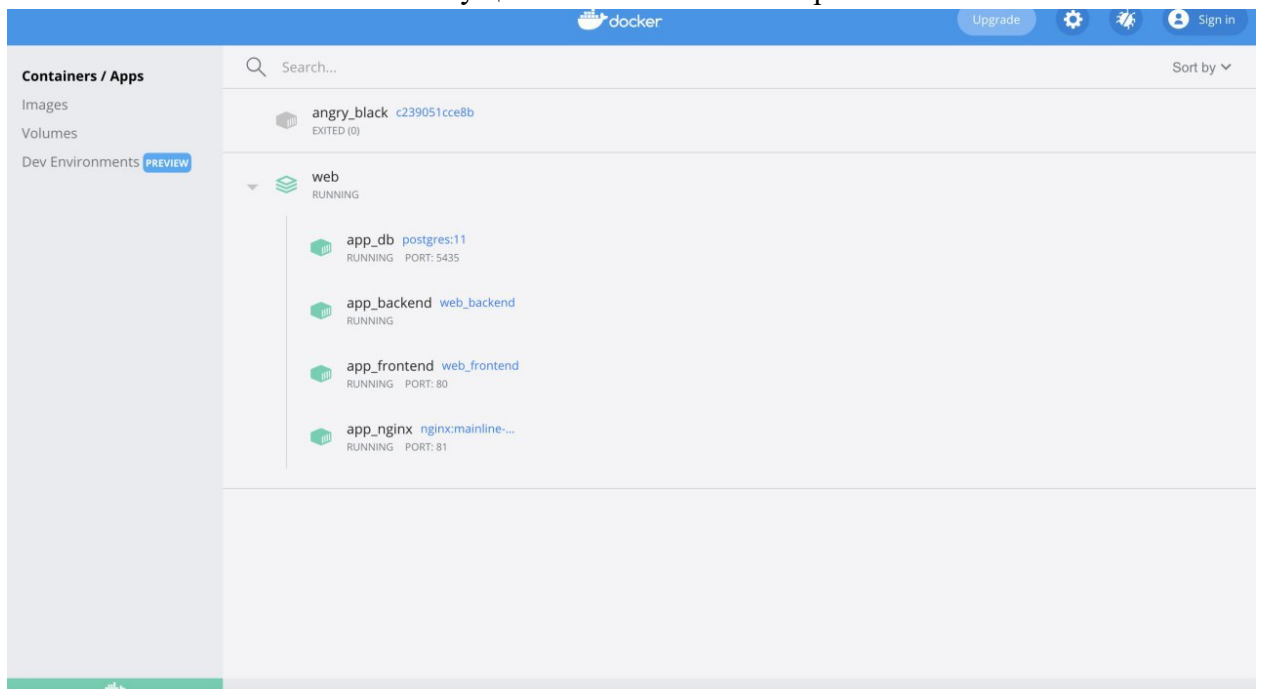


### 3. Серверное программирование

#### Выполнение запроса регистрации в postman



#### Запущенные docker контейнеры



## **docker-compose.yml**

```
version: '2.2'

services:
  backend:
    build:
      context: backend/
      dockerfile: Dockerfile
    container_name: app_backend
    command: uvicorn main:app --host 0.0.0.0 --port 81
    env_file:
      - ../.test.env
    depends_on:
      - db
    ports:
      - 81:81
    expose:
      - 81
    volumes:
      - static_volume:/home/app/static
    networks:
      - app_net

  db:
    image: postgres:11
    container_name: app_db
    volumes:
      - postgres_data:/var/lib/postgresql/data/
    env_file:
      - ../.test.env
    ports:
      - 5435:5432
    networks:
      - app_net

  nginx:
    image: nginx:mainline-alpine
    container_name: app_nginx
    volumes:
      - static_volume:/home/app/static
      - ../nginx:/etc/nginx/conf.d
    depends_on:
      - backend
      - frontend
    networks:
      - app_net

  frontend:
    build:
      context: web/
      dockerfile: Dockerfile
    container_name: app_frontend
    env_file:
      - ../.test.env
    depends_on:
```

```

        - backend
    ports:
        - 80:80
    expose:
        - 80
    volumes:
        - node_modules:/app/node_modules
    networks:
        - app_net

volumes:
    postgres_data:
    static_volume:
    node_modules:

networks:
    app_net:
driver: bridge

```

### nginx.conf

```

upstream backend {
    server backend:8000;
}
upstream frontend {
    server frontend:8080;
}
server {
    listen 81;
    server_name 127.0.0.1 localhost;
    client_max_body_size 100m;
    proxy_ignore_client_abort on;
    if_modified_since off;
    add_header Last-Modified "";
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header Host $host;
    proxy_redirect off;
    uwsgi_read_timeout 300s;
    location / {
        proxy_pass http://backend;
    }
}
server {
    listen 80;
    server_name 127.0.0.1 localhost;
    client_max_body_size 100m;
    if_modified_since off;
    add_header Last-Modified "";
    proxy_ignore_client_abort on;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header Host $host;
    proxy_redirect off;
    uwsgi_read_timeout 300s;
    location / {
        proxy_pass http://frontend;
    }
}

```

```
}
```

### **Dockerfile(frontend)**

```
FROM node:13.12.0-alpine as build
WORKDIR /app
ENV PATH /app/node_modules/.bin:$PATH
COPY package.json ./
COPY package-lock.json ./
RUN npm install --silent
RUN npm install react-scripts@3.4.1 -g --silent
COPY . ./
RUN npm run build

# production environment
FROM nginx:stable-alpine
COPY --from=build /app/build /usr/share/nginx/html
EXPOSE 80
CMD ["nginx", "-g", "daemon off;"]
```

### **Dockerfile(backend)**

```
FROM python:3.8
WORKDIR /home/app

ENV PYTHONDONTWRITEBYTECODE 1
ENV PYTHONUNBUFFERED 1

RUN pip install --upgrade pip

COPY ./requirements.txt /home/app/requirements.txt
RUN pip install -r requirements.txt

RUN apt-get update && apt-get install netcat -y

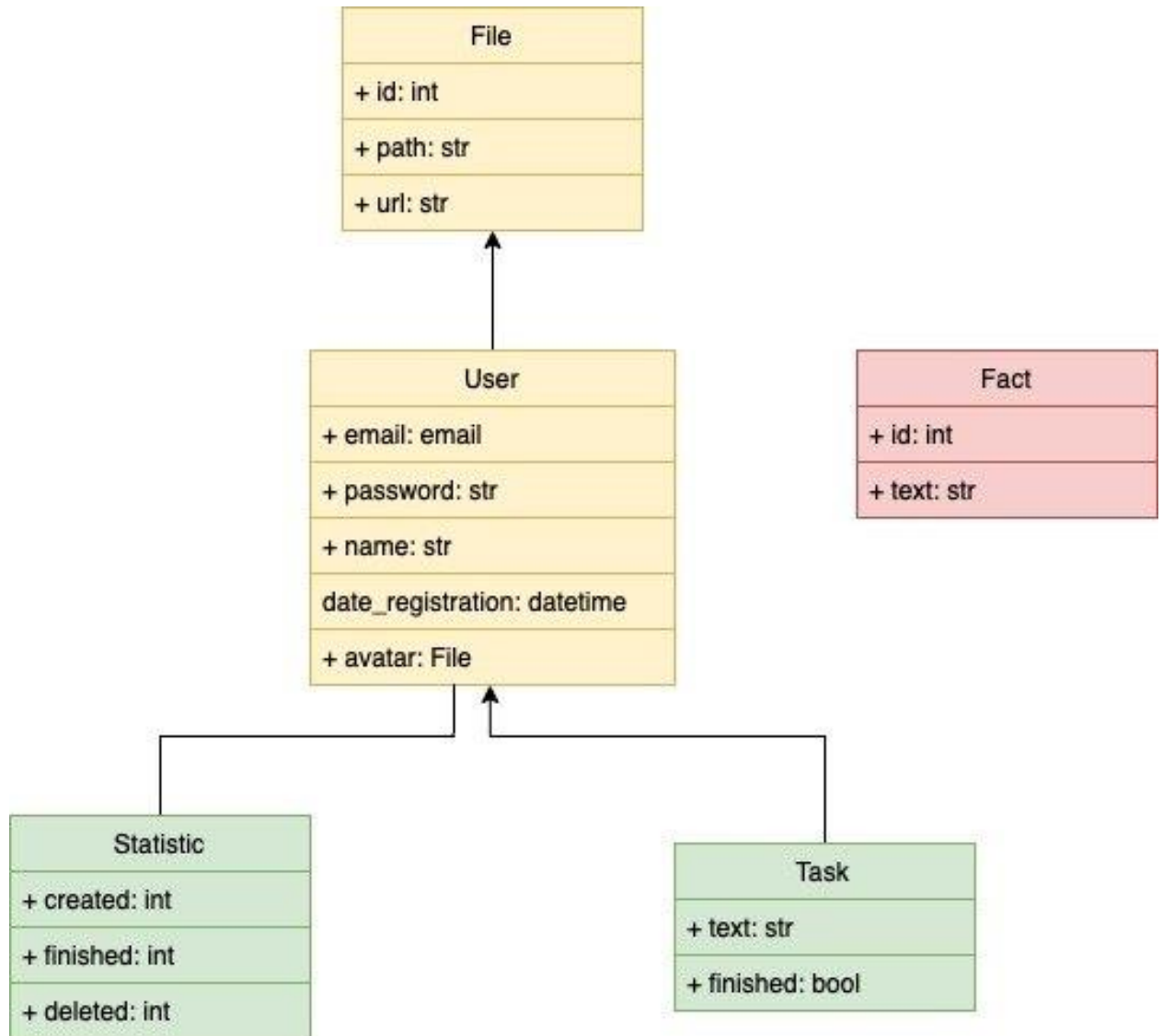
COPY . /home/app

RUN ["chmod", "+x", "/home/app/scripts/entrypoint.sh"]
ENTRYPOINT ["/home/app/scripts/entrypoint.sh"]
```

## 4. Разработка и проектирование базы данных

В качестве СУБД была выбрана Postgresql 11, в качестве ORM был использована асинхронная библиотека Tortoise

### Структура базы данных



### **Models.py**

```
from tortoise import models, fields, Tortoise

from config import DBPaths

Tortoise.init_models(DBPaths.base, 'models')

class File(models.Model):
    id = fields.BigIntField(pk=True)
    path = fields.CharField(max_length=250)

    class PydanticMeta:
        include = ('id', 'path')
```

### **schemas.py**

```
from pydantic import BaseModel
from tortoise.contrib.pydantic import PydanticModel, pydantic_model_creator
from db.base.models import File
from config import DOMAIN_BACKEND

class Status(BaseModel):
    message: str

class GetFile(PydanticModel):
    id: int
    url: str = None
    path: str

    def __init__(self, **kwargs):
        super(GetFile, self).__init__(**kwargs)
        self.url = f'{DOMAIN_BACKEND}/{self.path}'

    class Config:
        orig_model = File

class URL(PydanticModel):
    url: str
```

## 5. REST API

### Описание API

ToDo List

1.0

OAS3

/openapi.json

API

Authorize

Пользователи

^

GET

/users/permissions/

Get Users

✓

🔒

GET

/users/permissions/me

Get Me

✓

🔒

GET

/users/permissions/{user\_id}

Get User

✓

🔒

PUT

/users/permissions/set\_avatar/

Set Avatar

✓

🔒

POST

/users/auth/login

Login

✓

POST

/users/auth/logout

Logout

✓

🔒

POST

/users/auth/register

Register

✓

Задачи

^

GET

/tasks/statistic/

Get Statistic

✓

🔒

GET

/tasks/task/

Get All

✓

🔒

POST

/tasks/task/

Create

✓

🔒

PUT

/tasks/task/{task\_id}

Put

✓

🔒

DELETE

/tasks/task/{task\_id}

Delete

✓

🔒

Статистика

^

GET

/tasks/statistic/

Get Statistic

✓

🔒

Факты

^

GET

/facts

Get All

✓

POST

/facts

Create One

✓

DELETE

/facts

Delete All

✓

GET

/facts/{item\_id}

Get One

✓

PUT

/facts/{item\_id}

Update One

✓

DELETE

/facts/{item\_id}

Delete One

✓

GET

/facts/get\_random/

Get Random

✓

31

## Факт дня

GET /facts/get\_random/ Get Random

Parameters

No parameters

Execute Clear

Responses

Curl

```
curl -X 'GET' \
'http://127.0.0.1:8000/facts/get_random/' \
-H 'accept: application/json'
```

Request URL

```
http://127.0.0.1:8000/facts/get_random/
```

Server response

Code	Details
200	<p>Response body</p> <pre>{   "id": 3,   "text": "Микроволновая печь была изобретена, когда исследователь проходил мимо лучевой трубки, и плитка шоколада растаяла в его кармане." }</pre> <p>Response headers</p> <pre>content-length: 253 content-type: application/json date: Mon, 06 Dec 2021 17:31:20 GMT server: uvicorn</pre>

Responses

## Задачи

GET /tasks/task/ Get All

Parameters

No parameters

Execute Clear

Responses

Curl

```
curl -X 'GET' \
'http://127.0.0.1:8000/tasks/task/' \
-H 'accept: application/json'
```

Request URL

```
http://127.0.0.1:8000/tasks/task/
```

Server response

Code	Details
200	<p>Response body</p> <pre>[   {     "id": 4,     "text": "Дело 4",     "finished": false   },   {     "id": 3,     "text": "Дело 3",     "finished": false   },   {     "id": 2,     "text": "Дело 2",     "finished": true   } ]</pre>

Response headers



## 6. Работа с HTTP запросами

В качестве библиотеки для работы с аях запросами был выбран axios.

### api/tasks.js

```
import axios from "../index";
import {returnData} from "../utils";

const baseUrl = "/tasks"

export function getStatistic() {
  return axios.get(`${baseUrl}/statistic/`)
    .then(returnData)
}

export function getTaskList() {
  return axios.get(`${baseUrl}/task/`)
    .then(returnData)
}

export function createTask(text) {
  return axios.post(`${baseUrl}/task/`, {text})
    .then(returnData)
}

export function checkTask(taskId) {
  return axios.put(`${baseUrl}/task/${taskId}`)
    .then(returnData)
}

export function deleteTask(taskId) {
  return axios.delete(`${baseUrl}/task/${taskId}`)
    .then(returnData)
}
```

### api/auth.js

```
import axios from "../index";

const baseAuthURL = '/users/auth'

export function login({username, password}) {
  const bodyFormData = new FormData();

  bodyFormData.append('username', username)
  bodyFormData.append('password', password)
```

```

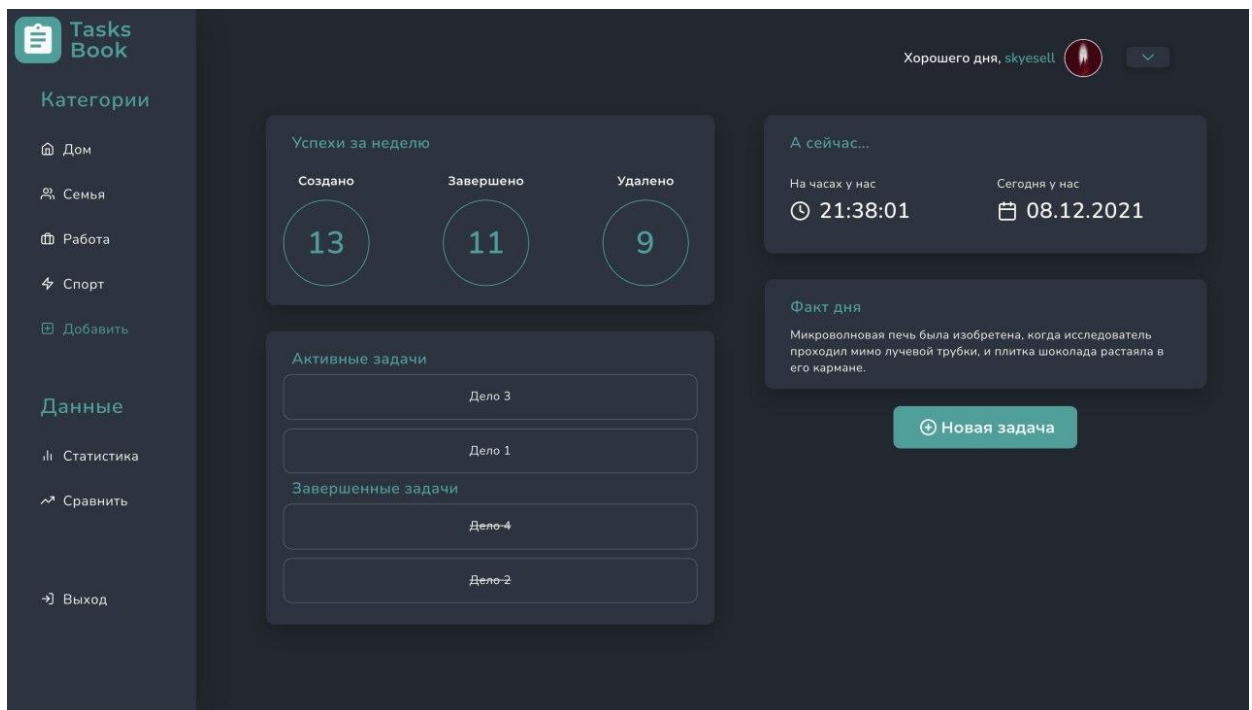
    return axios.post(`${baseAuthURL}/login`, bodyFormData)
  }

export function register(userData) {
  return axios.post(`${baseAuthURL}/register`, userData)
}

export function logout() {
  return axios.post(`${baseAuthURL}/logout`)
}

```

В результате было получено полностью функционирующее веб приложение



**Вывод:** В результате выполнения данной работы нам удалось создать полноценное веб-приложение, предназначенное для ведения списка дел. В ходе изучения мы стоклнулись с использованием HTML, CSS и Bootstrap. Написали клиентское приложение с использованием JavaScript и React. Создали REST API и спроектировали БД, используя в качестве СУБД PostgreSQL. А в конце связали бэкенд и фронтенд, взяв за основу библиотеку Axios.