

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Дисциплина:** Фронт-энд разработка

Отчет

Лабораторная работа 2: взаимодействие с внешним API

Выполнил:

Борсов Никита

Группа К33402

Проверил:

Добряков Д. И.

Санкт-Петербург

2023 г.

## Задача

Необходимо привязать верстку из первой лабораторной работы к внешнему API средствами fetch/axios/xhr, тем самым взаимодействуя из данными извне приложения.

## Ход работы

### 1. Подключение API Flask приложения

```
104
    Nikita B
105 @bp.route("/login2", methods=("GET", "POST"))
106 def login2():
107     if request.method == "POST":
108         username = request.form["username"]
109         password = request.form["password"]
110         db = get_db()
111         error = None
112         user = db.execute(
113             "SELECT * FROM user WHERE username = ?", (username,)
114         ).fetchone()
115
116         if user is None:
117             error = "Incorrect username."
118         elif not check_password_hash(user["password"], password):
119             error = "Incorrect password."
120
121     if error is None:
122         # store the user id in a new session and return to the index
123
124         session.clear()
125         session["user_id"] = user["id"]
126         return redirect(url_for("blog.index2"))
127
128     flash(error)
129
130     return render_template("auth/login2.html")
131
132
133 Nikita B
134 @bp.route("/logout")
135 def logout():
136     session.clear()
137     return redirect(url_for("auth.login2"))
```

```

122     cursor = conn.cursor()
123
124     if user_id == 0:
125         v3_data_1 = list(cursor.execute("SELECT initiate_timestamp FROM sessions WHERE conversation_start_timestamp IS NOT NULL"))
126         v3_data_2 = list(cursor.execute("SELECT conversation_start_timestamp FROM sessions WHERE conversation_start_timestamp IS NOT NULL"))
127     else:
128         v3_data_1 = list(cursor.execute("SELECT initiate_timestamp FROM sessions WHERE conversation_start_timestamp IS NOT NULL and court_id=?", (
129         v3_data_2 = list(cursor.execute("SELECT conversation_start_timestamp FROM sessions WHERE conversation_start_timestamp IS NOT NULL and court_id=?", (
130
131     v3_data_1 = ["".join(str(tup[0])) for tup in v3_data_1]
132     v3_data_2 = ["".join(str(tup[0])) for tup in v3_data_2]
133     v3_data_1 = [int(i) for i in v3_data_1]
134     v3_data_2 = [int(i) for i in v3_data_2]
135     v3_data_3 = [v3_data_2[i] - v3_data_1[i] for i in range(len(v3_data_1))]
136     v3 = f"{round(sum(v3_data_3) / len(v3_data_3) / 60, 2)}"
137
138     conn.commit()
139     conn.close()
140
141     return v3
142
143
144  Nikita B
145  def get_average_time_of_operator_dialogue(user_id):
146     conn = sqlite3.connect('../court-bot/database/db')
147     cursor = conn.cursor()
148
149     if user_id == 0:
150         v4_data_1 = list(cursor.execute("SELECT conversation_start_timestamp FROM sessions WHERE completion_timestamp IS NOT NULL"))
151         v4_data_2 = list(cursor.execute("SELECT completion_timestamp FROM sessions WHERE completion_timestamp IS NOT NULL"))
152     else:
153         v4_data_1 = list(cursor.execute("SELECT conversation_start_timestamp FROM sessions WHERE completion_timestamp IS NOT NULL and court_id=?", (
154         v4_data_2 = list(cursor.execute("SELECT completion_timestamp FROM sessions WHERE completion_timestamp IS NOT NULL and court_id=?", (user_id,
155
156     v4_data_1 = ["".join(str(tup[0])) for tup in v4_data_1]
157     v4_data_2 = ["".join(str(tup[0])) for tup in v4_data_2]
158     v4_data_1 = [int(i) for i in v4_data_1]
159     v4_data_2 = [int(i) for i in v4_data_2]

```

```

no usages  Nikita B
var randomChartData = function randomChartData(n, min, max) {
    var data = [];
    for (var i = 0; i < n; i++) {
        data.push(Math.floor(Math.random() * (max - min + 1)) + min);
    }

    return data;
};

5+ usages  Nikita B
function updateScale(chart) {
    let fromDateChartDisplay = document.getElementById("inputDateFrom").value;
    let toDateChartDisplay = document.getElementById("inputDateTo").value;
    console.log("FAC1!!!!", fromDateChartDisplay);

    if (fromDateChartDisplay !== "" && toDateChartDisplay !== "") {
        let left_filter_index = dates_sucs_sessions.indexOf(fromDateChartDisplay);
        let right_filter_index = dates_sucs_sessions.indexOf(toDateChartDisplay);

        dates_sucs_sessions_for_charts = dates_sucs_sessions.slice(left_filter_index, right_filter_index);
        points_sucs_sessions_for_charts = points_sucs_sessions.slice(left_filter_index, right_filter_index);
        dates_fail_sessions_for_charts = dates_fail_sessions.slice(left_filter_index, right_filter_index);
        points_fail_sessions_for_charts = points_fail_sessions.slice(left_filter_index, right_filter_index);
    }
}

```

## 2. Реализация фильтров

```

43
44 if (fromDateChartDisplay !== "" && toDateChartDisplay !== "") {
45     let left_filter_index = dates_sucs_sessions.indexOf(fromDateChartDisplay);
46     let right_filter_index = dates_sucs_sessions.indexOf(toDateChartDisplay);
47
48     dates_sucs_sessions_for_charts = dates_sucs_sessions.slice(left_filter_index, right_filter_index);
49     points_sucs_sessions_for_charts = points_sucs_sessions.slice(left_filter_index, right_filter_index);
50     dates_fail_sessions_for_charts = dates_fail_sessions.slice(left_filter_index, right_filter_index);
51     points_fail_sessions_for_charts = points_fail_sessions.slice(left_filter_index, right_filter_index);
52 }
53
54 if (chart === chrt1 || chart === chrt2) {
55     chart.data.datasets = [{
56         fill: false,
57         borderColor: chartColors["default"].primary,
58         borderWidth: 2,
59         borderDash: [],
60         borderDashOffset: 0.0,
61         pointBackgroundColor: chartColors["default"].primary,
62         pointBorderColor: 'rgba(255,255,255,0)',
63         pointHoverBackgroundColor: chartColors["default"].primary,
64         pointBorderWidth: 20,
65         pointHoverRadius: 4,
66         pointHoverBorderWidth: 15,
67         pointRadius: 4,
68         data: points_sucs_sessions_for_charts,
69     }, {
70         fill: false,
71         borderColor: chartColors["default"].info,
72         borderWidth: 2,
73         borderDash: [],
74         borderDashOffset: 0.0,
75         pointBackgroundColor: chartColors["default"].info,
76         pointBorderColor: 'rgba(255,255,255,0)',
77         pointHoverBackgroundColor: chartColors["default"].info,
78         pointBorderWidth: 20,
79         pointHoverRadius: 4,

```

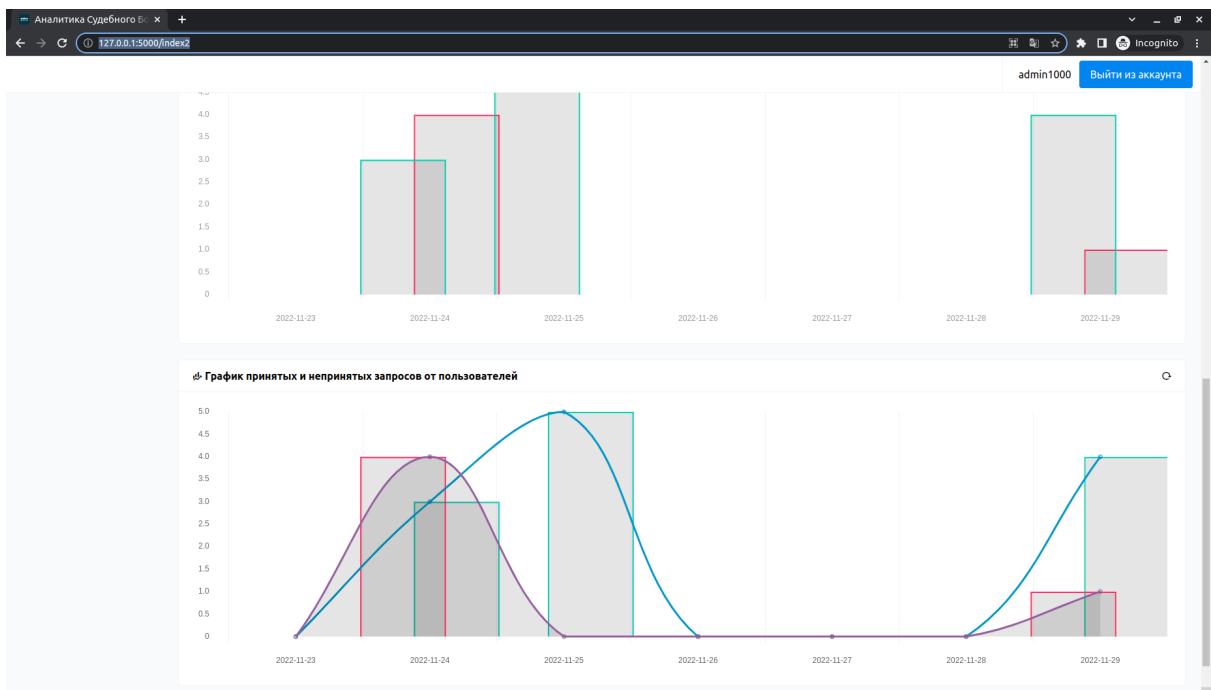
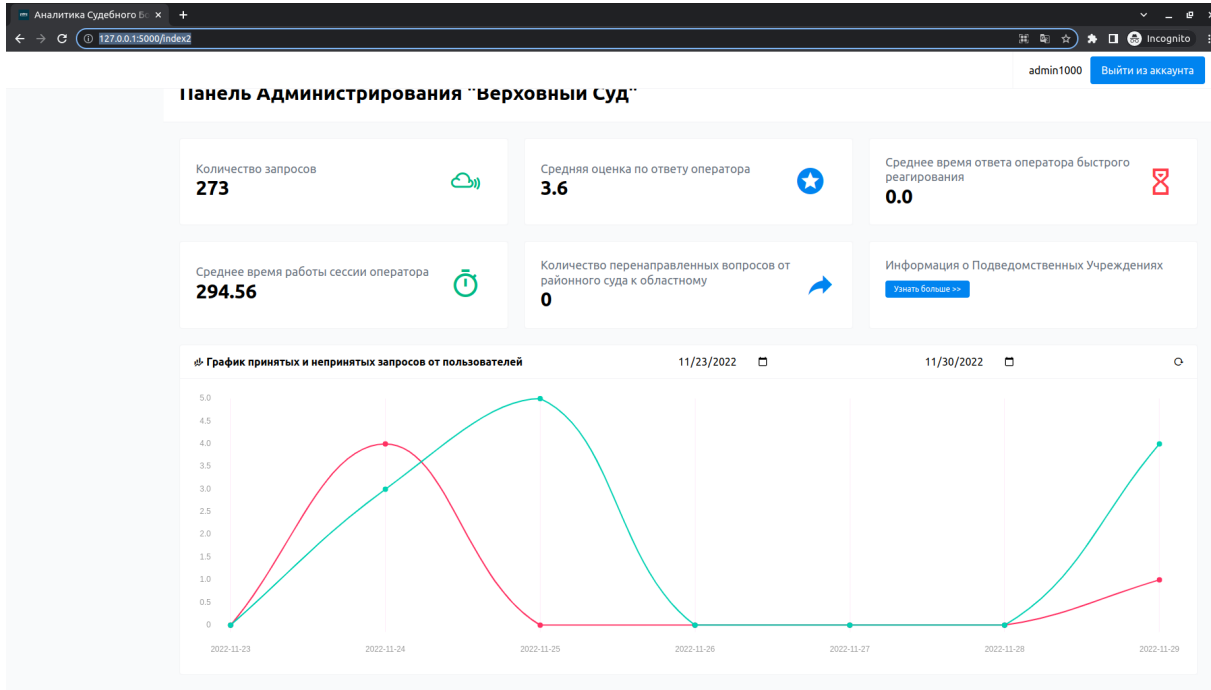
### 3. Итоговый результат

mm/dd/yyyy 📅

mm/dd/yyyy 📅

11/23/2022 📅

11/30/2022 📅



## **Вывод**

По итогу выполнения поставленной задачи - к имеющимся наработкам основанным на HTML, CSS и Bootstrap, был подключен и протестирован бэкэнд приложения по работе с внешним API который позволяет авторизовать пользователя, просматривать и редактировать данные.