



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н. Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н. Э. Баумана)

ФАКУЛЬТЕТ

«Информатика и системы управления»

КАФЕДРА

«Программное обеспечение ЭВМ и информационные технологии»

ОТЧЕТ

По лабораторной работе №3

По курсу: «Архитектура ЭВМ»

Студент:

Ле Ни Куанг

Группа:

ИУ7и-56Б

Преподаватель:

Попов А. Ю.

Москва

2020

Цель работы

Изучить AJAX Get, Post запросы, работать с шаблонизатором ознакомиться с файлами cookie и сессиями в NodeJs, укрепить знания html, css, javascript.

Задача 5

Задание 1

Создать сервер. Сервер должен выдавать страницу с тремя текстовыми полями и кнопкой. В поля ввода вбивается информация о почте, фамилии и номере телефона человека. При нажатии на кнопку "Отправить" введенная информация должна отправляться с помощью POST запроса на сервер и добавляться к концу файла (в файле накапливается информация). При этом на стороне сервера должна происходить проверка: являются ли почта и телефон уникальными. Если они уникальны, то идет добавление информации в файл. В противном случае добавление не происходит. При отправке ответа с сервера клиенту должно приходить сообщение с информацией о результате добавления (добавилось или не добавилось). Результат операции должен отображаться на странице.

Задание 2

Добавить серверу возможность отправлять клиенту ещё одну страницу. На данной странице должно быть поле ввода и кнопка. В поле ввода вводится почта человека. При нажатии на кнопку "Отправить" на сервер отправляется GET запрос. Сервер в ответ на GET запрос должен отправить информацию о человеке с данной почтой в формате JSON или сообщение об отсутствии человека с данной почтой.

Задание 3

Оформить внешний вид созданных страниц с помощью CSS. Информация со стилями CSS для каждой страницы должна храниться в отдельном файле. Стили CSS должны быть подключены к страницам.

Листинг 1: File index.js

```
1 const express = require('express')
2 const fs = require('fs')
3
4 const app = express()
5 const port = 5000
6
7 app.listen(port)
8 console.log('http://localhost:${port}/')
9
10
11 const way = __dirname + '/static'
12 app.use(express.static(way))
13
14 app.use((req, res, next) => {
15   res.header('Cache-Control', 'no-cache, no-store, must-revalidate')
16   res.header('Access-Control-Allow-Headers', 'Origin, X-Requested-With, Content-Type, Accept')
17   res.header('Access-Control-Allow-Origin', '*')
18   next()
19 })
20
21 function loadBody(req, callback) {
22   let body = []
23   req.on('data', (chunk) => {
24     body.push(chunk)
25   }).on('end', () => {
26     body = Buffer.concat(body).toString()
27     callback(body)
28   })
29 }
30
31 const dbfile = 'data/db.json'
32 let db = JSON.parse(fs.readFileSync(dbfile))
33
34 function isExist(n, e, p) {
35   for (let i in db) {
36     if (e === db[i].email || p === db[i].phone)
37       return true
38   }
39   return false
40 }
41
42 app.get('/data', (req, res) => {
43   res.sendFile(__dirname + '/' + dbfile)
44 })
45
46 app.post('/data', (req, res) => {
```

```

47     loadBody(req, (body) => {
48         let obj = JSON.parse(body)
49         if (isExist(...obj)) {
50             res.end('{"result":_false}')
51         } else {
52             let entry = {
53                 "name": obj[0],
54                 "email": obj[1],
55                 "phone": obj[2]
56             }
57             db.push(entry)
58             fs.writeFileSync(dbfile, JSON.stringify(db, null, 2))
59             res.end('{"result":_true}')
60         }
61     })
62 })
63
64 app.get('/find', (req, res) => {
65     const e = req.query.email
66     let result = {result: null}
67     for (let i in db) {
68         if (e === db[i].email) {
69             result = {result: db[i]}
70         }
71     }
72     res.end(JSON.stringify(result))
73 })

```

Листинг 2: File static/ajax.js

```

1 function ajaxGet(urlString, callback) {
2     let r = new XMLHttpRequest()
3     r.open('GET', urlString, true)
4     r.setRequestHeader('Content-Type', 'text/plain;charset=UTF-8')
5     r.send(null)
6     r.onload = function() {
7         callback(r.response)
8     }
9 }
10
11 function ajaxPost(urlString, bodyString, callback) {
12     let r = new XMLHttpRequest()
13     r.open('POST', urlString, true)
14     r.setRequestHeader('Content-Type', 'application/json;charset=UTF-8')
15     r.send(bodyString)
16     r.onload = function() {
17         callback(r.response)
18     }
19 }

```

Листинг 3: File static/index.html

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <meta charset="UTF-8">
5     <title>Lab 3-1</title>
6     <link rel="stylesheet" href="style/main.css">
7 </head>
8 <body>
9     <h1>Lab 3 - 1</h1><br><br>
10    <div class="card" style="width: 30%;>
11        <br><a href="1.html">Task 1</a><br><br>
12    </div>
13    <div class="card" style="width: 30%;>
14        <br><a href="2.html">Task 2</a><br><br>
15    </div>
16 </body>
17 </html>
```

Листинг 4: File static/1.html

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <meta charset="UTF-8">
5     <title>Lab 3-1 - Task 1</title>
6     <link rel="stylesheet" href="style/12.css">
7 </head>
8 <body>
9     <div id="form">
10        <p>Name</p>
11        <input id="name" type="text" minlength=3 required>
12
13        <p>Email</p>
14        <input id="email" type="email" autocomplete="off" required>
15
16        <p>Phone Number</p>
17        <input id="phone" type="tel" pattern="8[0-9]{10}"
18            minlength=11 maxlength=11 value="8" required>
19
20        <br><br>
21        <button onClick="post()">Submit</button>
22
23    </div>
24
25    <br><br>
26    <table id="TB" border="1px">
27        <tr>
```

```

28         <th>Name</th>
29         <th>Email</th>
30         <th>Phone number</th>
31     </tr>
32 </table>
33
34
35 <script src="/ajax.js"></script>
36 <script language="javascript">
37     let table = document.getElementById('TB')
38     let tbody = table.querySelector('tbody')
39     let name = document.getElementById('name')
40     let email = document.getElementById('email')
41     let phone = document.getElementById('phone')
42
43     function addRow(n, e, p) {
44         let tr = document.createElement('tr')
45         tr.innerHTML = '<td>${n}<td>${e}<td>${p}'
46         tbody.appendChild(tr)
47     }
48
49     function fetchData() {
50         ajaxGet('data', (listStr) => {
51             const list = JSON.parse(listStr)
52             for (let i in list) {
53                 addRow(...Object.values(list[i]))
54             }
55         })
56     }
57
58     window.onload = () => {
59         fetchData()
60     }
61
62     function post() {
63         if (name.checkValidity() &&
64             email.checkValidity() &&
65             phone.checkValidity()) {
66             ajaxPost('data', JSON.stringify([
67                 name.value, email.value, phone.value
68             ]), (answer) => {
69                 let result = JSON.parse(answer).result
70                 if (result) {
71                     addRow(name.value, email.value, phone.value)
72                     alert("Saved!")
73                 } else {
74                     alert("Email_or_phone_number_already_exist!")
75                 }
76             })
77         }
78     }

```

```

76         })
77     }
78 }
79 </script>
80
81 </body>
82 </html>

```

Листинг 5: File static/2.html

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4     <meta charset="UTF-8">
5     <title>Lab 3-1 - Task 2</title>
6     <link rel="stylesheet" href="style/12.css">
7 </head>
8 <body>
9     <p>Email</p>
10    <input id="email" type="email" autocomplete="off" required>
11    <br><br>
12    <button onclick="find()">Find</button>
13    <br><br>
14
15    <table id="TB" border="1px" style="width:80%">
16        <tr>
17            <th>Name</th>
18            <th>Email</th>
19            <th>Phone number</th>
20        </tr>
21        <tr>
22            <td id="lb-name"></td>
23            <td id="lb-email"></td>
24            <td id="lb-phone"></td>
25        </tr>
26    </table>
27
28    <script src="/ajax.js"></script>
29    <script language="javascript">
30        let email = document.getElementById('email')
31        let lbName = document.getElementById('lb-name')
32        let lbEmail = document.getElementById('lb-email')
33        let lbPhone = document.getElementById('lb-phone')
34
35        function find() {
36            if (email.checkValidity()) {
37                let url = 'find/?email=${email.value}'
38                ajaxGet(url, (answer) => {
39                    console.log(JSON.parse(answer))

```



```

40         let result = JSON.parse(answer).result
41         if (result) {
42             lbName.innerText = result.name
43             lbEmail.innerText = result.email
44             lbPhone.innerText = result.phone
45         } else {
46             alert("Not found!")
47         }
48     })
49 }
50 }
51 </script>
52
53 </body>
54 </html>

```

Листинг 6: File static/style/main.css

```

1 body {
2     font-family: Avenir, Helvetica, Arial, sans-serif;
3     -webkit-font-smoothing: antialiased;
4     -moz-osx-font-smoothing: grayscale;
5     text-align: center;
6     color: #2c3e50;
7     padding: 8px;
8     background-color: whitesmoke;
9     text-shadow: #66666660 2px 2px 10px;
10 }
11
12 .card {
13     border-radius: 6px;
14     margin-bottom: min(20px, 2vw);
15 }
16
17 .card:hover {
18     filter: brightness(105%);
19     box-shadow: #66666660 2px 2px 12px;
20 }
21
22 a {
23     color: #00796B;
24     text-decoration: none;
25     letter-spacing: 1px;
26 }
27
28 div {
29     margin: auto;
30 }

```

Листинг 7: File static/style/12.css

```
1 body {
2     font-family: Avenir, Helvetica, Arial, sans-serif;
3     -webkit-font-smoothing: antialiased;
4     -moz-osx-font-smoothing: grayscale;
5     text-align: center;
6     color: #2c3e50;
7     padding: 8px;
8     background-color: whitesmoke;
9     text-shadow: #66666660 2px 2px 10px;
10 }
11
12 table {
13     margin: auto;
14     border: 1px solid;
15     width: min(100%, 600px);
16     background-color: white;
17 }
18
19 th {
20     background-color: cornsilk;
21 }
22
23
24 button {
25     padding: 5px 10px 5px 10px;
26     cursor: pointer;
27     border-radius: 5px;
28     border: none;
29     background-color: lightblue;
30 }
31
32 button:hover {
33     filter: brightness(102%);
34     box-shadow: #66666660 2px 2px 10px;
35 }
```

Тест 1

Lab 3 - 1

Task 1

Task 2

Name

Email

Phone Number

Submit

Name	Email	Phone number
Horne	sherryhorne@blurrybus.com	89685453912
Daugherty	sherrydaugherty@blurrybus.com	88135682627
Haney	sherryhaney@blurrybus.com	88784942920
Rivas	sherryrivas@blurrybus.com	88575453954
Greer	sherrygreer@blurrybus.com	89195843211
Booker	sherrybooker@blurrybus.com	88924582907
Jarvis	sherryjarvis@blurrybus.com	89544512416
Brennan	sherrybrennan@blurrybus.com	88365413960

Name

Name

Email

test_email@gmail.com

Phone Number

80000000000

Submit

Name	Email	Phone number
Horne	sherryhorne@blurrybus.com	89685453912
Daugherty	sherrydaugherty@blurrybus.com	88135682627
Haney	sherryhaney@blurrybus.com	88784942920
Rivas	sherryrivas@blurrybus.com	88575453954
Greer	sherrygreer@blurrybus.com	89195843211
Booker	sherrybooker@blurrybus.com	88924582907
Jarvis	sherryjarvis@blurrybus.com	89544512416
Brennan	sherrybrennan@blurrybus.com	88365413960
Name	test_email@gmail.com	80000000000

localhost:5000 says

Email or phone number already exist!

OK

Email

test_email@gmail.com

Phone Number

80000000000

Submit

Name	Email	Phone number
Horne	sherryhorne@blurrybus.com	89685453912
Daugherty	sherrydaugherty@blurrybus.com	88135682627
Haney	sherryhaney@blurrybus.com	88784942920
Rivas	sherryrivas@blurrybus.com	88575453954
Greer	sherrygreer@blurrybus.com	89195843211
Booker	sherrybooker@blurrybus.com	88924582907
Jarvis	sherryjarvis@blurrybus.com	89544512416
Brennan	sherrybrennan@blurrybus.com	88365413960
Name	test_email@gmail.com	80000000000

Tect 2

Email

test_email@gmail.com

Find

Name	Email	Phone number
Name	test_email@gmail.com	80000000000

Задача 6

Задание 1

Создать сервер. В оперативной памяти на стороне сервера создать массив, в котором хранится информация о компьютерных играх (название игры, описание игры, возрастные ограничения). Создать страницу с помощью шаблонизатора. В url передаётся параметр возраст (целое число). Необходимо отображать на этой странице только те игры, у которых возрастное ограничение меньше, чем переданное в url значение.

Задание 2

Создать сервер. В оперативной памяти на стороне сервера создать массив, в котором хранится информация о пользователях (логин, пароль, хобби, возраст). На основе cookie реализовать авторизацию пользователей. Реализовать возможность для авторизованного пользователя просматривать информацию о себе.

Листинг 8: File index.js

```
1  'use strict'
2
3  const express = require('express')
4  const fs = require('fs')
5  const cookieSession = require('cookie-session')
6
7
8  const app = express()
9  const port = 5000
10
11 app.listen(port, () => {
12   console.log(`http://localhost:${port}/`)
13 })
14
15 app.set('view engine', 'hbs')
16
17
18 app.use((req, res, next) => {
19   res.header('Cache-Control', 'no-cache, no-store, must-revalidate')
20   res.header('Access-Control-Allow-Headers', 'Origin, X-Requested-With, Content-Type, Accept')
21   // res.header('Access-Control-Allow-Origin', '*')
22   next()
23 })
24
25
26 app.use(cookieSession({
27   name: 'session',
28   keys: ['hhh', 'qqq', 'vvv'],
29   maxAge: 24 * 60 * 60 * 7
30 }))
31
32
33 const games = JSON.parse(fs.readFileSync('data/game.json'))
34 const accounts = JSON.parse(fs.readFileSync('data/account.json'))
35
36 let session = {}
37
38
39 app.get('/', (req, res) => {
40   res.sendFile(__dirname + '/static/index.html')
41 })
42
43
44 app.get('/game', (req, res) => {
45   const age = req.query.age
46   const obj = {
```

```

47         filtered: games.filter((e) => e.age <= age)
48     }
49     res.render('game.hbs', obj)
50 })
51
52 function findAccount(e, p) {
53     if (!e || !p) return -1
54     let res = -1
55     for (let i in accounts) {
56         if (e === accounts[i].email && p === accounts[i].password) {
57             res = i
58             break
59         }
60     }
61     return res
62 }
63
64
65 app.get('/account', (req, res) => {
66     let sessionID = req.session.sessionID
67
68     if (session[sessionID] !== undefined) {
69         res.render('profile.hbs', accounts[session[sessionID]])
70         return
71     }
72
73     const e = req.query.email
74     const p = req.query.password
75     const index = parseInt(findAccount(e, p))
76
77     if (index === -1) {
78         res.sendFile(__dirname + '/static/login.html')
79     } else {
80         sessionID = Math.random().toString(36).substring(2, 15)
81         session[sessionID] = index
82         req.session.sessionID = sessionID
83         res.redirect('/account')
84         // res.render('profile.hbs', accounts[index])
85     }
86 })
87
88
89 app.get('/account/logout', (req, res) => {
90     delete session[req.session.sessionID]
91     req.session = null
92     res.redirect('/account')
93     // res.sendFile(__dirname + '/static/login.html')
94 })

```


Листинг 9: File func.js

```
1 export function sum() {
2     let s = 0
3     for (let i of arguments) {
4         s += i
5     }
6     return s
7 }
8
9 export function f4(a, b, c) {
10     let arr = []
11     for (let i = a; i <= b; i++) {
12         if (i % c === 0)
13             arr.push(i)
14     }
15     return arr
16 }
```

Листинг 10: File static/index.html

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <meta charset="UTF-8">
5     <title>Lab 3-2</title>
6 </head>
7 <body>
8     <h1>Lab 3 - 2</h1><br><br>
9     <div class="card" style="width: 30%;">
10         <br><a href="/game">Task 1</a><br><br>
11     </div>
12     <div class="card" style="width: 30%;">
13         <br><a href="/account">Task 2</a><br><br>
14     </div>
15 </body>
16 <style>
17 body {
18     font-family: Avenir, Helvetica, Arial, sans-serif;
19     -webkit-font-smoothing: antialiased;
20     -moz-osx-font-smoothing: grayscale;
21     text-align: center;
22     color: #2c3e50;
23     padding: 8px;
24     background-color: whitesmoke;
25     text-shadow: #66666660 2px 2px 10px;
26 }
27
28 .card {
```

```

29     border-radius: 6px;
30     margin-bottom: min(20px, 2vw);
31 }
32
33 .card:hover {
34     filter: brightness(102%);
35     box-shadow: #66666660 2px 2px 12px;
36 }
37
38 a {
39     color: #00796B;
40     text-decoration: none;
41     letter-spacing: 1px;
42     word-spacing: 1px;
43 }
44
45 div {
46     margin: auto;
47 }
48 </style>
49 </html>

```

Листинг 11: File static/login.html

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4     <meta charset="UTF-8">
5     <title>Login</title>
6 </head>
7 <body>
8     <form>
9         <table>
10             <tr>
11                 <th>Email</th>
12                 <td><input name="email" type="email" required></td>
13             </tr>
14             <tr>
15                 <th>Password</th>
16                 <td><input name="password" type="password" required></td>
17             </tr>
18         </table>
19         <input id="submit" type="submit" value="Login"></input>
20     </form>
21 </body>
22 <style>
23     body {
24         font-family: Avenir, Helvetica, Arial, sans-serif;
25         -webkit-font-smoothing: antialiased;

```

```

26     -moz-osx-font-smoothing: grayscale;
27     text-align: center;
28     color: #2c3e50;
29     padding: 8px;
30     background-color: whitesmoke;
31     text-shadow: #66666660 2px 2px 10px;
32 }
33
34 table {
35     margin: 50px auto 20px auto;
36     width: min(60%, 400px);
37 }
38
39 #submit {
40     padding: 5px 10px 5px 10px;
41     cursor: pointer;
42     border-radius: 5px;
43     border: none;
44     background-color: #c0f0ff;
45     font-size: medium;
46 }
47
48 #submit:hover {
49     filter: brightness(102%);
50     box-shadow: #66666660 2px 2px 10px;
51 }
52
53 input {
54     margin: 5px;
55 }
56
57 </style>
58 </html>

```

Листинг 12: File views/game.hbs

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4     <meta charset="UTF-8">
5     <title>Game</title>
6 </head>
7 <body>
8
9 <form method="GET" action="/game">
10     <label>Filter by age</label>
11     <input name="age" type="number" min=0 max=100 value=10 required>
12     <input id="submit" type="submit" value="Filter"></input>
13 </form>

```

```

14
15 <br><br>
16 <table border="1px">
17     <tr>
18         <th>Game</th>
19         <th>Age Restriction</th>
20         <th>Detail</th>
21     </tr>
22     {{#each filtered}}
23     <tr class="row">
24         <td>{{this.name}}</td>
25         <td>{{this.age}}</td>
26         <td>{{this.about}}</td>
27     </tr>
28     {{/each}}
29 </table>
30
31
32 </body>
33 <style>
34     body {
35         font-family: Avenir, Helvetica, Arial, sans-serif;
36         -webkit-font-smoothing: antialiased;
37         -moz-osx-font-smoothing: grayscale;
38         text-align: center;
39         color: #2c3e50;
40         padding: 8px;
41         background-color: whitesmoke;
42         text-shadow: #66666660 2px 2px 10px;
43     }
44
45     table {
46         margin: auto;
47         border: 1px solid;
48         width: min(100%, 600px);
49         background-color: white;
50     }
51
52     th {
53         background-color: cornsilk;
54     }
55
56     #submit {
57         padding: 5px 10px 5px 10px;
58         cursor: pointer;
59         border-radius: 5px;
60         border: none;
61         background-color: #c0f0ff;

```

```

62 }
63
64 #submit:hover {
65     filter: brightness(102%);
66     box-shadow: #66666660 2px 2px 10px;
67 }
68
69 input {
70     margin: 10px;
71     padding: 2px 0 2px 0;
72 }
73
74 td,th {
75     border: none;
76 }
77
78 .row {
79     border: none;
80     margin-bottom: min(20px, 2vw);
81 }
82
83 .row:hover {
84     filter: brightness(105%);
85     background-color: #c0f0ff;
86 }
87 </style>
88 </html>

```

Листинг 13: File views/profile.hbs

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4     <meta charset="UTF-8">
5     <title>Profile</title>
6 </head>
7 <body>
8 <table border="1px">
9     <tr>
10         <th>Name:</th>
11         <td>{{name}}</td>
12     </tr>
13     <tr>
14         <th>Age:</th>
15         <td>{{age}}</td>
16     </tr>
17     <tr>
18         <th>Hobbies:</th>
19         <td>

```

```

20         {{#each hobbies}}
21             {{this}},
22         {{/each}}
23     </td>
24 </tr>
25 </table>
26 <form method="GET" action="/account/logout">
27     <input id="submit" type="submit" value="Logout"></input>
28 </form>
29 </body>
30 <style>
31     body {
32         font-family: Avenir, Helvetica, Arial, sans-serif;
33         -webkit-font-smoothing: antialiased;
34         -moz-osx-font-smoothing: grayscale;
35         text-align: center;
36         color: #2c3e50;
37         padding: 8px;
38         background-color: whitesmoke;
39         text-shadow: #66666660 2px 2px 10px;
40     }
41
42     table {
43         margin: 50px auto 20px auto;
44         border: 1px solid;
45         width: min(60%, 400px);
46         background-color: white;
47     }
48
49     #submit {
50         padding: 5px 10px 5px 10px;
51         cursor: pointer;
52         border-radius: 5px;
53         border: none;
54         background-color: #c0f0ff;
55     }
56
57     #submit:hover {
58         filter: brightness(102%);
59         box-shadow: #66666660 2px 2px 10px;
60     }
61
62     td,th {
63         border: none;
64     }
65 </style>
66 </html>

```

Tect 1

Filter by age Filter

Game	Age Restriction	Detail
dolor	4	Qui elit do in sit do Lorem non occaecat nulla duis.
excepteur	8	Proident veniam dolore ad sint adipisicing anim et.
dolore	4	Magna velit esse occaecat in elit sint eiusmod fugiat et veniam.
nostrud	2	Lorem minim et non ipsum id.
dolore	3	Nisi consectetur laboris dolore ex anim.
magna	4	Lorem duis sint sint pariatur est consequat consequat adipisicing pariatur irure.
excepteur	5	Est deserunt exercitation ea pariatur.
quis	2	Proident reprehenderit eiusmod deserunt labore aliquip nulla in aliquip quis.
quis	5	Elit commodo labore tempor Lorem ut reprehenderit commodo ea nulla mollit cupidatat exercitation.
mollit	0	Veniam laboris sint consequat commodo adipisicing sunt sit.
ea	1	Enim irure aute do nisi labore quis qui ipsum nisi dolore proident sint ut.
Lorem	4	Proident consectetur mollit esse velit incididunt quis incididunt.
anim	6	Velit elit reprehenderit ea sint irure ut nisi sit.
dolor	5	Culpa voluptate cillum ea consectetur ullamco.
cupidatat	1	Est dolore laboris elit qui ad non Lorem.
sit	1	Veniam ea sit Lorem minim do tempor culpa deserunt sunt reprehenderit exercitation aute pariatur sint.
dolor	0	Esse nisi ipsum pariatur eu nulla.
enim	5	Dolore adipisicing amet deserunt veniam veniam exercitation et.
non	6	Dolore elit eiusmod elit labore do nostrud magna minim deserunt eiusmod adipisicing ad ullamco.

Tect 2

Email

johnconner@musanpoly.com

Password

Login

Name:

John Conner

Age:

14

Hobbies:

consequat, laborum, mollit,

Logout

Вывод

При выполнении лабораторной работы было изучено AJAX Get, Post запросы, работал с шаблонизатором ознакомился с файлами cookie и сеансами в NodeJs, укрепил знания html, css, javascript.