Финальный проект (Оля)

Документация к финальному проекту

Ссылка на задачу	JUS-234
Статус страницы	В работе
Описание	Документация к сайту Вank Saint-Petersburg

- Фичи, к которым идут требования
 - 1. Экран ввода логина и пароля demo и demo
 - 2. Экран ввода номера для смс: номер *1111. Код из смс 0000
 - 3. Экран СЧЕТА, раздел "Выписки"
- Описание требований
 - 1. Длина логина > 8 и <20 символов, только латинские буквы и цифры.
 - 1.1. Пароль > 8 и <20 символов, только латинские буквы, цифры и спец. символы. Как минимум, две латинские буквы должны быть в верхнем регистре. Как минимум должен быть один спец. символ.
 - 1.2 При правильных данных и нажатии на кнопку "Войти" осуществляется переход к окну подтверждения по смс.
 - 1.3 При 3 неудачных попытках пройти авторизацию, аккаунт блокируется.
 - 2. Номер для смс должен быть верифицирован в настройках аккаунта.
 - 2.1 Код подтверждения смс состоит из 4 знаков и содержит только цифры.
 - 2.2 Возможность повторной отправки кода осуществляется каждые 60 секунд.
 - 2.3 При верном коде и нажатие кнопки "Войти" должен осуществляться переход на главную страницу аккаунта.
 - 2.4 При 3 неудачных попытках входа должна происходить блокировка аккаунта.
 - 3. При нажатии на вкладку "ВАЛЮТА" текст цвет названия вкладки меняется с черного на красный.
 - 3.1 Должно быть всего три вкладки "Простой режим", "Биржевой мост" и "Мой курс".
 - 3.2 По умолчанию всегда открывается вкладка "Простой режим".
 - 3.3 По умолчанию в поле "Счет списания" подтягивается рублевый счет клиента, а в поле "Счет зачисления" счет в USD. Если условие для умолчания не подходит, то поля по умолчанию не заполняются.
 - 3.4 Для всех счетов используется маска (разделение пробелами между разрядами):

XXXXX XXX X XXXX XXXXXXX

▼ Test-cases & Bug-reports

Тест-кейсы

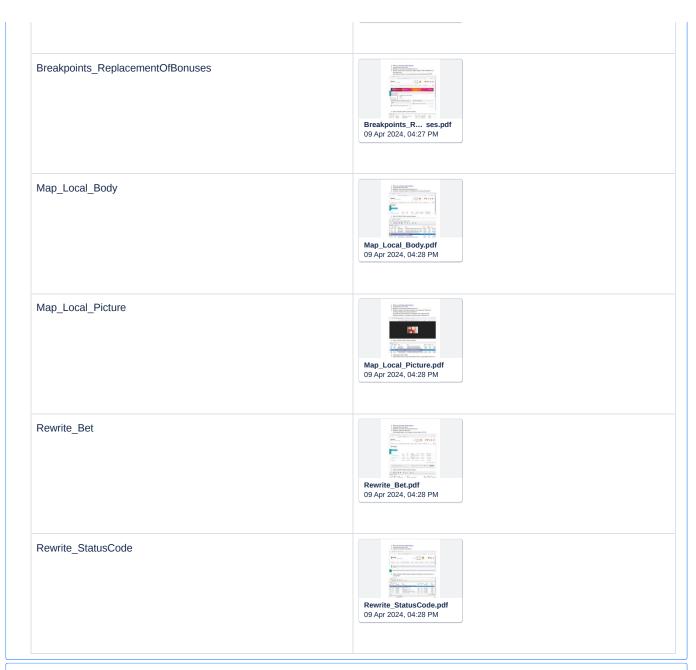


Блокировка карты	Blocking the card	Blocking the card.pdf 03 Apr 2024, 05:42 PM
Открытие нового вклада	Opening a new deposit	Opening a new sit.pdf 03 Apr 2024, 05:42 PM
Создание запроса на перевод денег с карты	Creating a request to transfer money from a card	Creating a requ ard.pdf 03 Apr 2024, 05:42 PM

🛕 Баг-репорт

Описание	Баг-репорт	PDF
Возможность разблокировать карту после вечной блокировки по причине Потеряна/Украдена	■ JUS-393: Возможность разблокироват ь карту после вечной блокировки по при чине Потеряна/Украдена роме	Unlocking the card.pdf 03 Apr 2024, 05:35 PM
Не сохраняется информация по "Общему лимиту" для карты	■ JUS-371: Не сохраняется информаци я по "Общему лимиту" для карты воме	Total card limit.pdf 03 Apr 2024, 05:35 PM
При открытии страницы "Валюта" появляется надпись "Биржевой мост - Страница не найдена"	■ JUS-370: При открытии страницы "Ва люта" появляется надпись "Биржевой мо ст - Страница не найдена". роме	Currency Page.pdf 03 Apr 2024, 05:36 PM

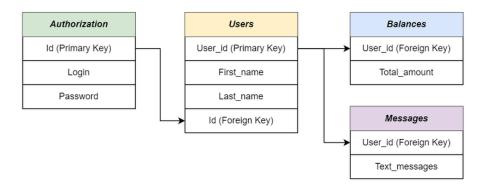
Breakpoints_Headers Breakpoints_he...ers.pdf 09 Apr 2024, 04:27 PM



▼ SQL DataBases - Bank SPB



DataBase BANK-SPB



Запросы

```
1 CREATE DATABASE BSPB;
 2
 3 USE BSPB;
 4
5 CREATE TABLE Authorization (
      -> Id INT (10) PRIMARY KEY AUTO_INCREMENT,
      -> Login VARCHAR(20) NOT NULL,
7
8
      -> Password VARCHAR(20) NOT NULL);
9
10 INSERT INTO Authorization (login, password) VALUES ('Di', 'qwerty'), ('Gizmo', 'HiWorld8'), ('Les', '12345678')
11 INSERT INTO Authorization (login, password) VALUES ('Gavrik','Mat*er7P'), ('Sheva','Qua6qwe8'), ('Losyash','
12
13 CREATE TABLE Users (
14
15
      -> User_id INT PRIMARY KEY AUTO_INCREMENT,
16
17
       -> First_name VARCHAR(30) NOT NULL,
18
19
      -> Last_name VARCHAR(30) NOT NULL,
20
      -> Email VARCHAR(50) NOT NULL,
21
23
       -> Phone INT(10) NOT NULL,
24
25
       -> Id INT, FOREIGN KEY Users (Id) REFERENCES Authorization (Id) );
26
27 INSERT INTO Users (First_name, Last_name, Email, Phone, Id) VALUES ('Olya', 'Demidova', 'demidova_o@gmail.com
28 INSERT INTO Users (First_name, Last_name, Email, Phone, Id) VALUES ('Valentina', 'Gavrilova', 'gavrilova_v@gr
29
30 CREATE TABLE Balances (
31
     -> User_id INT,
      -> Total_amount INT NOT NULL,
32
33
       -> FOREIGN KEY Balances (User_id) REFERENCES Users (User_id));
34
35 INSERT INTO Balances (User_id, Total_amount) VALUES (1, 250000), (2, 789000), (3, 962300);
36 INSERT INTO Balances (User_id, Total_amount) VALUES (13, 1200000), (14, 257000), (15, 30000);
```

```
37
38 CREATE TABLE Messages (
39
      -> User_id INT,
      -> Id_messages INT,
40
41
      -> Date_messages DATE,
      -> Text_message TEXT (500),
43
       -> FOREIGN KEY Messages (User_id) REFERENCES Users (User_id));
44
45 INSERT INTO Messages (User_id, Id_messages, Date_messages, Text_message) VALUES (1,0001,'2024-04-04','Hello v
46 INSERT INTO Messages (User_id, Id_messages, Date_messages, Text_message) VALUES (13,0001,'2024-04-06','I am v
47
48
49 SELECT * FROM Messages;
50 SELECT * FROM Authorization;
51 SELECT * FROM balances;
52 SELECT * FROM users;
```

```
qwerty
HiWorld8
12345678
Gizmo
Les
Gavrik
```

```
Olya
| Jenya
| Sergey
| Valentina
| Aleksandr
| Aleksey
                                                                                                                    demidova_o@gmail.com
sharapov_j@gmail.com
lesanov_s@gmail.com
gavrilova_v@gmail.com
shevalkin_a@gmail.com
losyash_a@gmail.com
                                                                Demidova
                                                              Demidova
Sharapov
Lesanov
Gavrilova
Shevalkin
Losyash
                                                                                                                                                                                                                                  55325
55326
55478
55535
55203
55535
```

```
sql> SELECT * FROM balances;
User id | Total amount
rows in set (0.00 sec)
```

```
ysql> SELECT * FROM Messages;
User_id | Id_messages | Date_messages | Text_message
                                                                       Hello world!!!
My name is Olya!
My friend!
Go home
I am tired
I am very cool girl
When will lunch be?
I am hungry
How much time is left until the end of the working day?
                                           2024-04-04
2024-04-04
                                           2024-04-01
2024-02-21
2024-02-22
                                           2024-04-06
2024-04-05
2024-04-05
                                           2024-02-21
    ws in set (0.00 sec)
```

▼ Bash_script

```
3 echo "Программа по созданию, переименованию и удалению файлов"
 4 echo "Введите путь к директории:"
 5 read dir_path
 6
7 есһо "Введите имя первого файла"
8 read file1
9
10 touch "${dir_path}/${file1}"
11
12 echo "Введите имя второго файла"
13 read file2
14
15 touch "${dir_path}/${file2}"
16
17 echo "Переименовываем первый файл ${file1}"
18 есно "Введите новое имя первого файла"
19 read new_file1
20 mv "${dir_path}/${file1}" "${dir_path}/${new_file1}"
21
22 echo "Копируем файл ${file2}"
23 echo "Введите новое имя копии второго файла"
24 read new_file2
25 cp "${dir_path}/${file2}" "${dir_path}/${new_file2}"
27 echo "Удаляем оригинальный второй файл ${file2}"
28 rm "${dir_path}/${file2}"
29
30 есho "Операции завершены"
```

→ Bash_command

а. Перейти в домашнюю директорию

cd

b. Создать новую папку mkdir papka_test (имя папки)

c. Перейти в нее cd papka_test

d. Создать в ней новый файл touch file.txt (имя файла)

e. Перейти в текстовый редактор и написать о ваших впечатлениях от курса на данный момент, что хорошо, что плохо и тд vim file.txt

Press "i"

All good!!!

I feel like I'm getting smarter!!!

f. Выйти и сохранить этот текст

Esc -> Shift+: -> wq

g. Посмотреть конкретное словосочетание в файле grep 'I feel' file.txt

h. Посмотреть конкретное словосочетание в файле и перенаправить поток в новый файл grep 'I feel' file.txt >> file2.txt

і. Посмотреть, что находится в вашей директории ls

j. Как посмотреть, что находится в вашей директории, НО, включая скрытые файлы ls -a

k. Посмотреть, где вы находитесь pwd

I. Вы в домашней директории > создать 2 папки > перейти в первую любую папку > создать там файл > скопировать и перенести этот файл во вторую папку

mkdir papka_test papka_test2

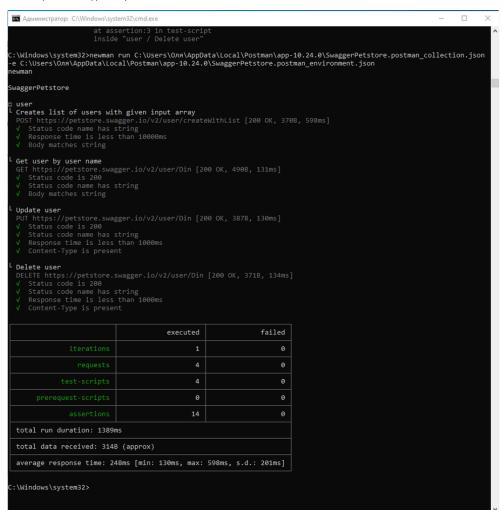
cd papka_test

touch file.txt

cp -r file.txt C:/Users/Оля/papka_test2



▼ Postman - Запуск коллекции в командной строке



```
→ Postman - Collection

     1 {
            "info": {
     2
     3
                "_postman_id": "d6c9ff23-dc98-4484-b600-ba8cd81d6fa2",
     4
                "name": "SwaggerPetstore",
     5
                "schema": "https://schema.getpostman.com/json/collection/v2.1.0/collection.json",
                "_exporter_id": "31897740"
     6
     7
            },
            "item": [
     8
     9
                {
                     "name": "user",
    10
                    "item": [
    11
    12
                         {
                             "name": "Creates list of users with given input array",
    13
    14
                             "event": [
    15
                                 {
                                     "listen": "test",
    16
                                     "script": {
    17
    18
                                         "exec": [
    19
                                             "pm.test(\"Status code name has string\", function () \{\r",
    20
                                                pm.response.to.have.status(\"OK\");\r",
                                             "});\r",
    21
                                             "\r",
    22
    23
                                             "pm.test(\"Response time is less than 10000ms\", function () \{\r",
                                                  pm.expect(pm.response.responseTime).to.be.below(10000);\r",
    24
    25
                                             "});\r",
    26
                                             "\r",
                                             "\r",
    27
    28
                                             "pm.test(\"Body matches string\", function () {\r",
                                                  pm.expect(pm.response.text()).to.include(\"code\",\"type\",\"messa
    29
                                             "});"
    30
    31
                                         ],
                                         "type": "text/javascript",
    32
                                         "packages": {}
    33
    34
                                     }
                                }
    35
                             ],
    36
                             "request": {
    37
                                 "method": "POST",
    38
    39
                                 "header": [],
                                 "body": {
    40
    41
                                     "mode": "raw",
                                     "raw": "[\r\n \"id\": 0,\r\n \"username\": \"Din\",\r\n
    42
                                                                                                             \"first!
                                     "options": {
    43
    44
                                         "raw": {
                                             "language": "json"
    45
    46
                                         }
    47
                                     }
                                },
    48
                                 "url": {
    49
    50
                                     "raw": "{{BASE_URL}}/user/createWithList",
                                     "host": [
    51
    52
                                         "{{BASE_URL}}"
    53
                                     ],
    54
                                     "path": [
    55
                                         "user",
```

```
"createWithList"
 56
 57
                                 ]
 58
                             }
                         },
 59
                          "response": []
 60
 61
                     },
 62
                     {
 63
                          "name": "Get user by user name",
 64
                         "event": [
                             {
 65
                                  "listen": "test",
 66
                                  "script": {
 67
                                      "exec": [
 68
 69
                                          "pm.test(\"Status code is 200\", function () {\r",
 70
                                             pm.response.to.have.status(200);\r",
 71
                                          "});\r",
                                          "\r",
 72
                                          "pm.test(\"Status code name has string\", function () \{\r",
 73
 74
                                          " pm.response.to.have.status(\"OK\");\r",
                                          "});\r",
 75
                                          "\r",
 76
 77
                                          "\r",
 78
                                          "pm.test(\"Body matches string\", function () \{\r",
 79
                                               pm.expect(pm.response.text()).to.include(\"username\",\"firstName\"
                                          "});"
                                      ],
 81
 82
                                      "type": "text/javascript",
                                      "packages": {}
 83
 84
                                  }
 85
                              }
                         ],
 86
                          "request": {
 87
 88
                              "method": "GET",
                              "header": [],
 89
                              "url": {
 90
                                  "raw": "{{BASE_URL}}/user/Din",
 91
                                  "host": [
 92
                                     "{{BASE_URL}}"
 93
 94
                                  "path": [
 95
 96
                                      "user",
                                      "Din"
 97
 98
                                  ]
 99
                             }
                         },
100
101
                         "response": []
102
                     },
103
                          "name": "Update user",
104
                         "event": [
105
106
                              {
                                  "listen": "test",
107
                                  "script": {
108
109
                                      "exec": [
                                          "pm.test(\"Status code is 200\", function () \{\r",
110
                                               pm.response.to.have.status(200);\r",
111
112
                                          "});\r",
                                          "\r",
113
```

```
"pm.test(\"Status code name has string\", function () {\r",
114
115
                                             pm.response.to.have.status(\"OK\");\r",
                                          "});\r",
116
                                          "\r",
117
                                          "pm.test(\"Response time is less than 1000ms\", function () \{\r",
118
119
                                               pm.expect(pm.response.responseTime).to.be.below(1000);\r",
                                          "});\r",
120
121
                                          "\r",
                                          "pm.test(\"Content-Type is present\", function () {\r",
122
                                             pm.response.to.have.header(\"Content-Type\");\r",
123
                                          "});\r",
124
125
126
                                      ],
                                      "type": "text/javascript",
128
                                      "packages": {}
129
                                 }
130
                              }
131
                         ],
132
                         "request": {
                              "method": "PUT",
133
                              "header": [],
134
                              "body": {
                                  "mode": "raw",
136
137
                                  "raw": "{\r\n \"id\": 0,\r\n \"username\": \"Din\",\r\n \"firstName\": \"Dmi
138
                                  "options": {
                                      "raw": {
139
140
                                          "language": "json"
141
142
                                  }
                             },
                              "url": {
144
                                  "raw": "{{BASE_URL}}/user/Din",
145
                                  "host": [
                                      "{{BASE_URL}}"
147
148
                                  ],
149
                                  "path": [
150
                                      "user",
                                      "Din"
151
                                  ]
152
153
                             }
                         },
                          "response": []
155
156
                     },
157
                     {
                          "name": "Delete user",
158
159
                          "event": [
160
                             {
                                  "listen": "test",
161
162
                                  "script": {
                                      "exec": [
163
164
                                          "pm.test(\"Status code is 200\", function () \{\r",
165
                                               pm.response.to.have.status(200);\r",
                                          "});\r",
166
167
                                          "pm.test(\"Status code name has string\", function () \{\r",
168
                                               pm.response.to.have.status(\"OK\");\r",
169
170
                                          "});\r",
                                          "\r",
171
```

```
"pm.test(\"Response time is less than 1000ms\", function () {\r",
172
                                              pm.expect(pm.response.responseTime).to.be.below(1000);\r",
173
174
                                          "});\r",
                                          "\r",
175
                                          "pm.test(\"Content-Type is present\", function () {\r",
176
177
                                             pm.response.to.have.header(\"Content-Type\");\r",
                                          "});\r",
178
179
180
                                      ],
                                      "type": "text/javascript",
181
182
                                      "packages": {}
183
                                 }
                             }
184
                         ],
                         "request": {
186
187
                             "method": "DELETE",
                              "header": [],
188
189
                              "url": {
190
                                 "raw": "{{BASE_URL}}/user/Din",
                                  "host": [
191
                                     "{{BASE_URL}}"
192
                                  ],
                                  "path": [
194
195
                                      "user",
                                     "Din"
                                  ]
197
198
                             }
199
                         },
200
                         "response": []
                 ],
202
203
                 "description": "Operation about user"
205
         ]
206 }
```

```
1 {
 2
        "id": "d6dfbdff-2c77-428a-b2d4-d55e5d49ec1e",
 3
        "name": "SwaggerPetstore",
        "values": [
 4
 5
           {
 6
                "key": "BASE_URL",
 7
               "value": "https://petstore.swagger.io/v2",
                "type": "default",
 8
 9
                "enabled": true
10
           }
11
12
        "_postman_variable_scope": "environment",
13
        "_postman_exported_at": "2024-04-02T18:40:37.143Z",
        "_postman_exported_using": "Postman/10.24.1"
14
15 }
```

▼ Postman - Environment

