**Автоматическая запись к врачу с получением уведомлений (Quick appointment with a doctor)**

*Новокшанов Евгений Андреевич  
г. Москва, ГБОУ Физматшкола No2007, 11 класс*

QAWD - это приложение, которое сможет заметно улучшить вашу жизнь в сфере записи к врачу. Если вы стоите в онлайн очередях или очередях поликлиник, то с QAWD вы сможете освободить свое время и посвятить его любимым занятиям и хобби, потому что мое веб-приложение будет выполнять запись к врачу без непосредственного наблюдения пользователя за его работой. Вам остается только нажать на кнопку и получить уведомление на почту о успешной записи.

Цель работы:

Написание программы для автоматической записи к врачу с получением уведомлений.

Задачи:

1) решить проблему потери времени людей, которые стоят в онлайн-очередях для записи на прием к врачу;

2) облегчить процесс записи к врачу на определенное время и дату.

Методы и инструменты:

1) программа написана на языке Python 3.7;

2) использована библиотека Selenium WebDriver (библиотека для управления браузерами);

3) работа с веб-сервисами.

Описание:

После введения определенных параметров записи (интервалы даты и времени, специализация врача, Ф.И.О. врача) выполняется автоматический поиск и отслеживание доступного для записи к врачу даты и времени. После того, как программа проведет запись, она отправит уведомление с указанием даты и времени приема у врач.

Содержание:  
1) Введение

2) Используемые методы

3) Интерфейс

4) Связующая программа

5) Основная программа

6) Отправка сообщений

7) Заключение

8) Приложение

Введение

Люди давно уже не стоят часами в очередях, чтобы записаться к врачу. Существует множество сервисов, через которые можно выполнить запись, не выходя из дома, однако и часто требует много времени. Люди, имеющие онкологические и сердечно-сосудистые заболевания, не могут записаться на прием, т.к. все время занято, и они вынуждены постоянно следить, не освободилось ли место в нужное время.

Я решил разобраться с этой проблемой и написать собственное веб-приложение, которое сможет автоматически искать свободное время для записи. Это освободило бы много времени у людей, которые в этом нуждаются.

Основание моего приложения состоит из следующих идей:

1. Быстрая запись
2. Если нет подходящего времени, то программа продолжает поиск
3. Обеспечение скорости записи
4. Понятный и простой интерфейс, которым будет удобно пользоваться не только людям среднего возраста , но и людям преклонного возраста.

page219image2145442656

Я считаю, что именно веб-приложение является наиболее удобным в использовании, т.к. оно не занимает памяти на устройстве, и пользоваться им можно где угодно и когда угодно.

В чем же заключаются отличия моего приложения от аналогов в Google Play, App Store и сайтов гос. услуг? Оно заключается в том, что пользователю не нужно будет заходить непосредственно на сайт в установленное приложение, если необходимо будет лишь открыть вкладку в браузере по ссылке и вписать свои данные, необходимые при записи. Далее программа выполнит всю работу сама и отправит вам сообщение о успешной записи на почту.

Моя программа ничего не весит, поэтом у не занимает место на устройстве, а также предоставляет больший функционал, чем аналоги.

1 Используемые методы

Приложение было написано на языке Python 3. Также для создания веб-интерфейса использовался язык HTML5.

Я выбрал язык Python, потому что он прост в использовании и для него написано множество библиотек, которые упрощают работу.

Приложение состоит из нескольких функциональных блоков:

1. Программа, которая осуществляет запись
2. HTML страница
3. Программа, которая связывает HTML страницу и программу записи к врачу, используя данные, введённые пользователем.
4. Программа, которая отвечает за отправку сообщения об успешной записи к врачу.

Ниже представлена связь блоков между собой:

HTML

Связка

Основа

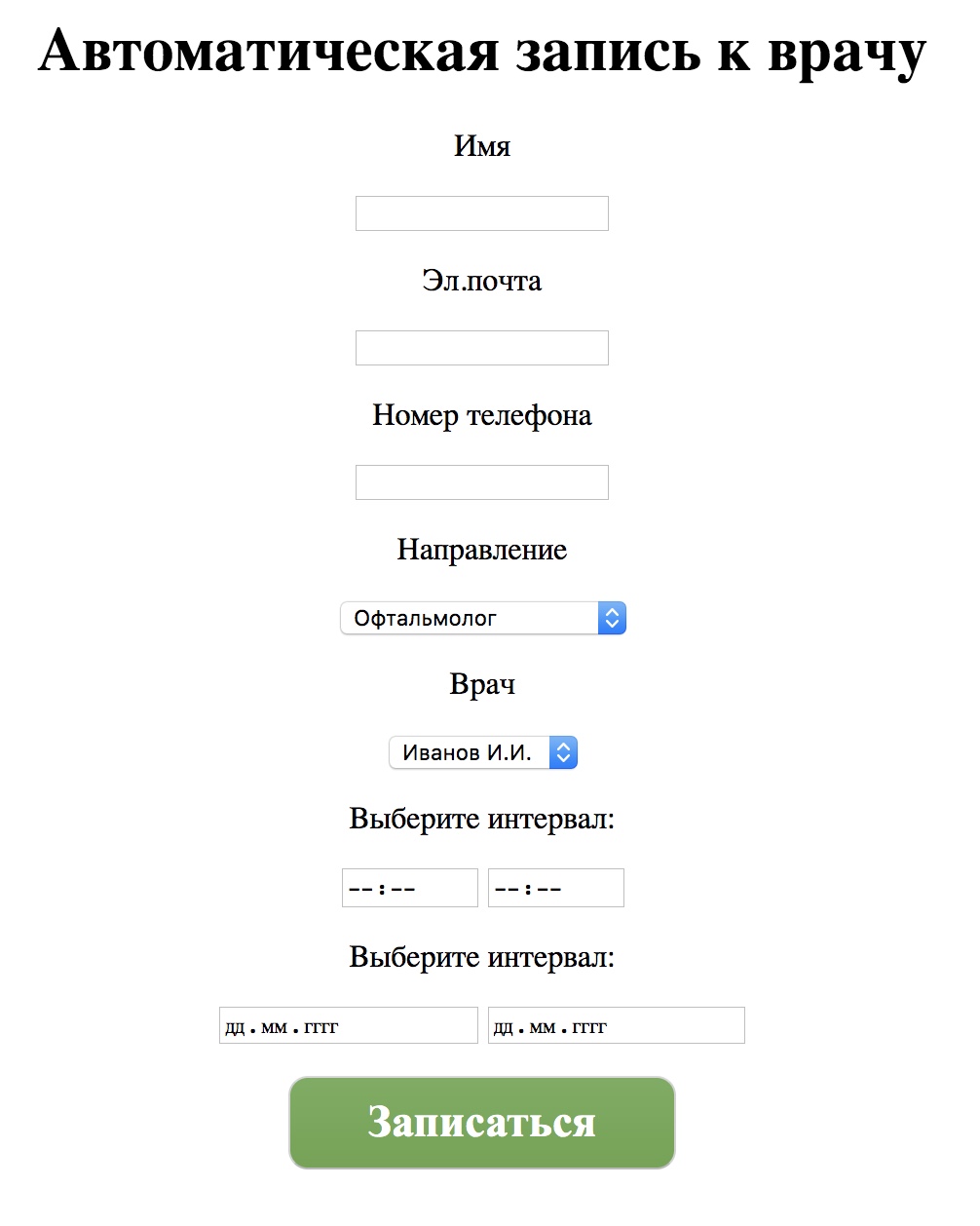
Сообщение

2 Интерфейс (HTML страница)

При переходе по ссылке на веб-приложение, вы увидите название веб-приложения и строки, необходимые для заполнения:

1. Имя
2. Электронная почта
3. Номер телефона
4. Направление (например, Офтальмолог)
5. Врач
6. Интервал времени, в которое вам удобно посетить врача
7. Интервал дней, в которые вам необходимо посетить врача

После того, как вы заполните все выше перчисленное, вам необходимо нажать на кнопку «Записаться»

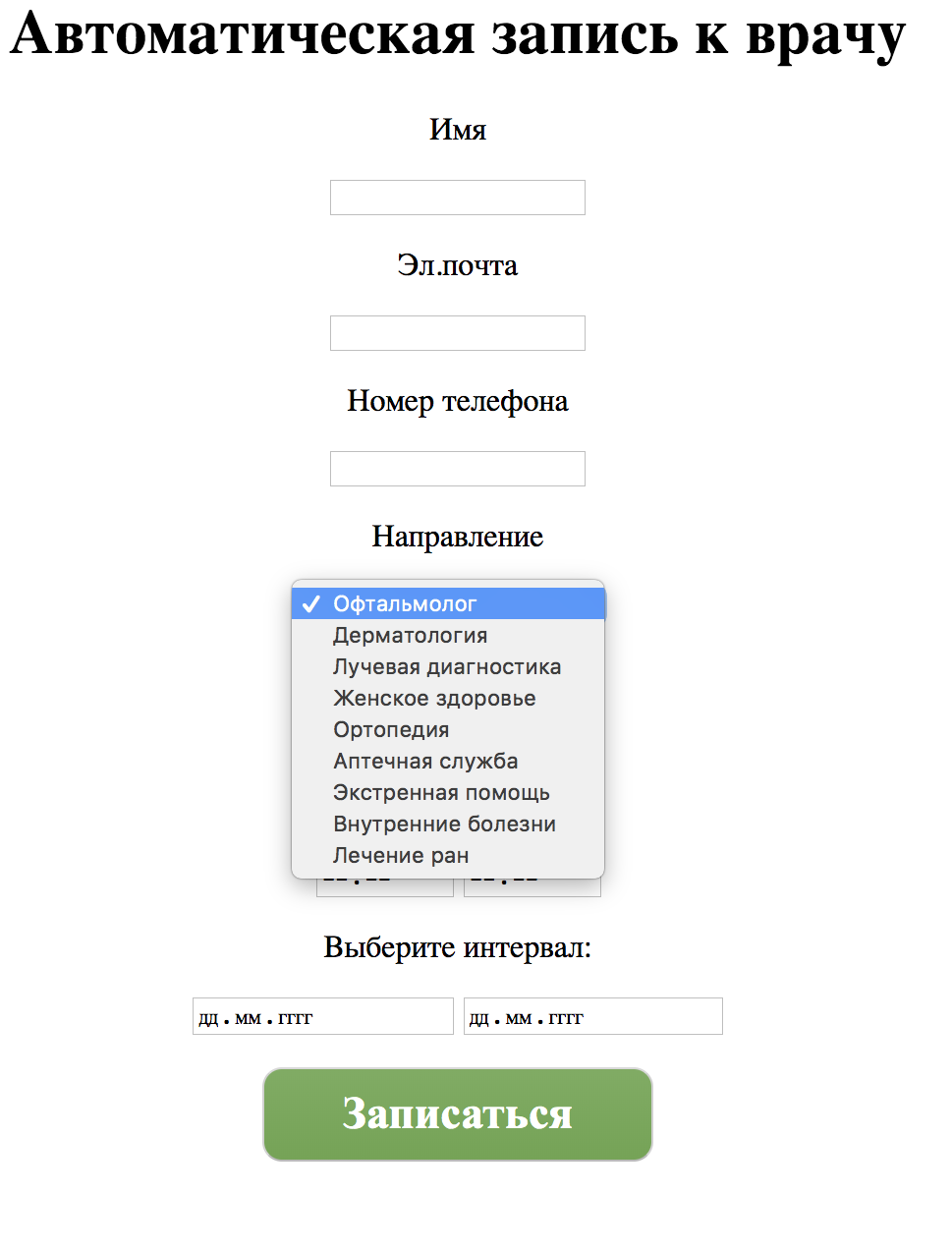


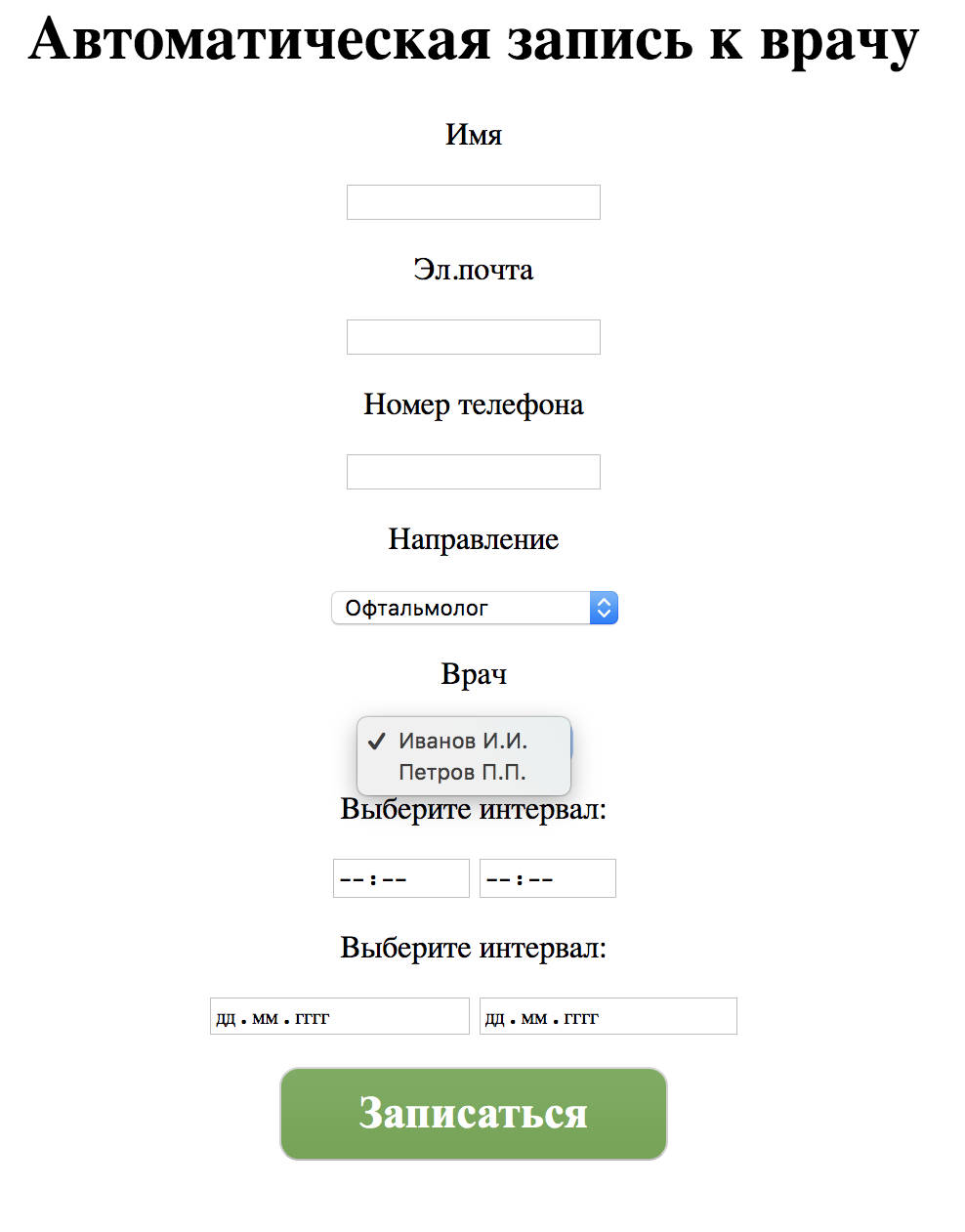
Для того, что бы это сделать, мне необходимо было изучить работу HTML5.

После нажатия на эту кнопку, данные пользователя передаются в связующую программу.

Элемент страницы с тегом style отвечает за раскраску кнопки «Записаться».

При выборе направления и врача открывается окно с дня выбора направления и доктора соответственно.





А вот и код страницы:

3 Связующая программа

Связующая программа, несмотря на свое название выполняет одну из самых важных функций.

Эта программа считывает данные, которые пользователь вводил на странице. После эти данные отправляются в основную программу. Именно в этой программе осущестляется постоянный мониторинг страницы записи. Это необходимо для того, чтобы пользователь мог изменить данные времени и даты. Для этого использовалась библиотека Flask.

4 Основная программа:

В этой программе осущестляется запись к врачу по данным, которые были переданы Связующей программой.

Все действия выполняются благодаря библиотеке Selenium. Она позволяет работать с веб-интерфейсом.

Программа делится на несколько функций:

1. Функция выбора напрвления
2. Функция выбора доктора
3. Функция выбора подходящего времени
4. Функция выбора подходящей даты
5. Функции, которые отвечают за регистрацию
6. Функции, отвечающие за активацию вышеперечисленных

Все функци реализованы путем кликов

Поэтому путе ссылок на определенные «кнопки» мы их нажимаем.

Функция pecialty\_selection(a) отвечает за выбор направления. После перехода на соответствующую страницу сайта(<https://zhenyanovokshanov1.wixsite.com/klinika/blank-5>) функция начнет проверку «кнопок» на соответствие с данным направлением.

Функция doctor\_choice(target\_name) ищет на сайте необходимого доктора с помощью метода XPATH по тексту “doctor\_name”

Функции data\_choice и time\_choce работаю вместе:

Функция data\_choice выбирает дату и проверяет, есть ли свободное время в этот день путем вызова функции time\_choce. Именно поэтому в эту функцию передается 4 переменных:

-интервал дат data\_start, data\_finish

-интервал времени time\_start, time\_finish

Так же это обеспечивает «гибкость» программы, поэтому ее можно будет изменить для других сайтов. Я писал программу для сайта, который был специально для этого создан, что бы не нарушать работу реальных сайтов и клиник.

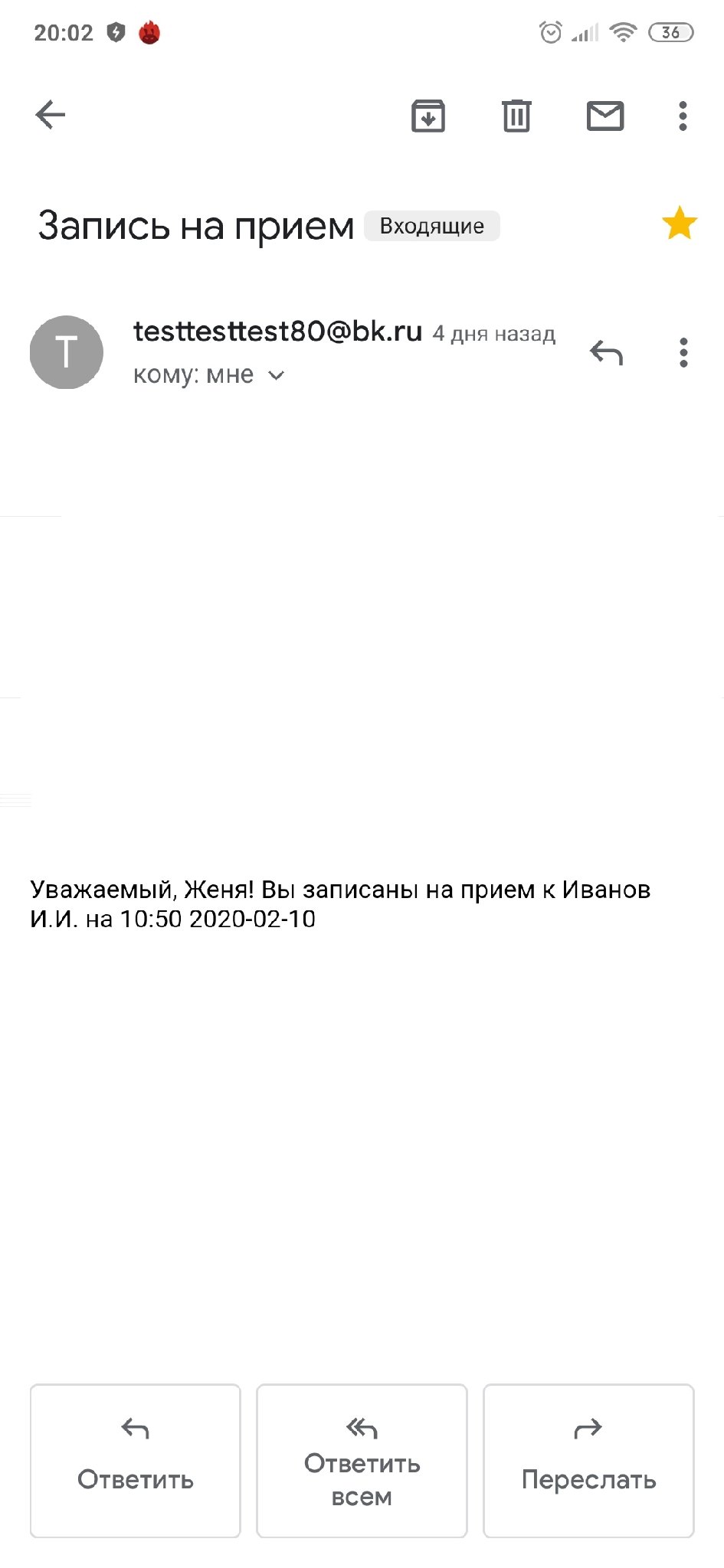
Самую главную роль в этом блоке выполняет последняя функция, которая обеспечивает связь между Связующей программой и функциями, которые отвечают за запись. Так же эта функция отправляет данные в программу отправки сообщения.

5 Отправка сообщений:

Данная часть программы предназначена, для уведомления пользователя об успешной записи. На почту указанную пользователем отправляется сообщение в данной форме: **"Уважаемый, {**user**}! Вы записаны на прием к {**doctor**} на {**time**} {**day**}"**

Поэтому пользователь будет не толькло предупрежден о записи, но и узнает дату, время и доктора.

Поиск нужной кнопки осщестляется путем поиска элементов по XPATH: By.XPATH, **f'\*//h3[contains(text(), "{**a**}")]'**













Заключение :

За время написание программы я не только хорошо проводил время, но и изучал то, что раньше для меня было непонятным. Я впервые работал с HTML кодом страницы сайта клиники и интерфейса моего веб-приложения.



Список использованной литературы

1. 1  SurfaceView [Электронный ресурс] -

http://developer.alexanderklimov.ru/android/views/surfaceview.ph

p

1. 2  Список с множественным выбором и собственным адаптером ArrayAdapter [Электронный ресурс] - http://developer.alexanderklimov.ru/android/listview/multiselect\_c ustomarrayadapter.php
2. 3  Диалоговое окно AlertDialog [Электронный ресурс] - http://developer.alexanderklimov.ru/android/alertdialog.php
3. 4  OkHttp [Электронный ресурс] - http://square.github.io/okhttp/
4. 5  Android OkHttp Example [Электронный ресурс] -

http://www.zoftino.com/android-okhttp-example

1. 6  Работа с JSON [Электронный ресурс] - http://java-

help.ru/android-json/

1. 7  JSON: основы использования [Электронный ресурс] - https://ruseller.com/lessons.php?id=1212&rub=28
2. 8  Библиотека GSON [Электронный ресурс] - http://developer.alexanderklimov.ru/android/library/gson.php
3. 9  Glide [Электронный ресурс] - https://github.com/bumptech/glide
4. 10  Android Glide Example [Электронный ресурс] -

https://www.dev2qa.com/android-glide-example/

1. 11  Матричные фильтры обработки изображений [Электронный

ресурс] - https://habr.com/ru/post/142818/

1. 12  Графические фильтры на основе матрицы скручивания

[Электронный ресурс] - http://wb0.ru/articles/120.htm

Приложение:

Интерфейс(HTML код):

<!DOCTYPE **html**>  
<**html lang="en"**>  
<**head**>  
 <**meta charset="UTF-8"**>  
 <**title**>Login</**title**>  
 <**style**>  
 .text {  
 text-align: center;  
 }  
 .myButton  
 {  
 box-shadow:inset 0px 0px 0px 0px #c4ccc0;  
 background:linear-gradient(to bottom, #74ad5a 5%, #68a54b 100%);  
 background-color:#74ad5a;  
 border-radius:10px;  
 display:inline-block;  
 cursor:pointer;  
 color:#ffffff;  
 font-family:Times New Roman;  
 font-size:23px;  
 font-weight:bold;  
 padding:10px 40px;  
 text-decoration:none;  
 }  
 .btn  
 {  
 /\* по-умолчанию для <button>, но пригодится для <a> \*/  
 display: inline-block;  
 text-align: center;  
 text-decoration: none;  
 /\* создаём маленькие отступы, если кнопки перенесутся на две строки \*/  
 margin: 2px 0;  
 /\* невидимая граница (понадобится для цвета при наведении/фокусе) \*/  
 border: solid 1px transparent;  
 border-radius: 4px;  
 /\* размер строится из текста и отступов (без width/height) \*/  
 padding: 0.5em 1em;  
 /\* убедитесь, что достаточно контраста! \*/  
 color: #ffffff;  
 background-color: #9555af;  
 }  
 </**style**>  
</**head**>  
<**body**>  
 <**div class="text"**>  
 <**form action="" method="post"**>  
  
 <**h1**>Автоматическая запись к врачу</**h1**>  
 <**p**>  
 <**label for="username"**>Имя</**label**>  
 </**p**>  
 <**p**>  
 <**input type="text" name="username"**>  
 </**p**>  
  
  
 <**p**>  
 <**label for="email"**>Эл.почта</**label**>  
 </**p**>  
 <**p**>  
 <**input type="text" name="email"**>  
 </**p**>  
  
 <**p**>  
 <**label for="phone\_number"**>Номер телефона</**label**>  
 </**p**>  
 <**p**>  
 <**input type="text" name="phone\_number"**>  
 </**p**>  
  
  
  
  
 <**p**>  
 <**label for="direction"**>Направление</**label**>  
 </**p**>  
  
 <**select name="direction"**>  
 <**option value="direction\_1"**>Офтальмолог</**option**>  
  
 <**option value="direction\_2"**>Дерматология</**option**>  
  
 <**option value="direction\_3"**>Лучевая диагностика</**option**>  
  
 <**option value="direction\_4"**>Женское здоровье</**option**>  
  
 <**option value="direction\_5"**>Ортопедия</**option**>  
  
 <**option value="direction\_6"**>Аптечная служба</**option**>  
  
 <**option value="direction\_7"**>Экстренная помощь</**option**>  
  
 <**option value="direction\_8"**>Внутренние болезни</**option**>  
  
 <**option value="direction\_9"**>Лечение ран</**option**>  
  
  
 </**select**>  
 <**p**>  
 <**label for="doctor"**>Врач</**label**>  
 </**p**>  
 <**select name="doctor"**>  
 <**option value="doctor\_1"**>Иванов И.И.</**option**>  
  
 <**option value="doctor\_2"**>Петров П.П.</**option**>  
  
  
  
  
 </**select**>  
 <**p**>  
 Выберите интервал:  
 </**p**>  
 <**p**>  
 <**input type="time" name="time1"**>  
 <**input type="time" name="time2"**>  
  
 </**p**>  
 <**p**>  
 Выберите интервал:  
 </**p**>  
 <**p**>  
 <**input type="date" name="calendar1"**>  
 <**input type="date" name="calendar2"**>  
 </**p**>  
  
  
 <**button class="myButton" type="submit"**>Записаться</**button**>  
  
 </**form**>  
 </**div**>  
</**body**>

</**html**>

Связующая программа:

**from** flask **import** Flask, render\_template, request  
**from** Klinika\_way **import** take\_zapis  
**from** threading **import** Thread  
**from** selenium **import** webdriver  
**from** selenium.webdriver.support.ui **import** WebDriverWait  
**from** selenium.webdriver.support **import** expected\_conditions **as** EC  
**from** selenium.webdriver.common.by **import** By  
**from** webdriver\_manager.chrome **import** ChromeDriverManager  
**import** time  
  
app = Flask(\_\_name\_\_)  
  
a = **'Офтальмолог'**target\_name = **'Иванов И.И.'**direction\_elem = [**'Офтальмолог'**, **'Дерматология'**, **'Лучевая диагностика'**, **'Женское здоровье'**, **'Ортопедия'**, **'Аптечная служба'**, **'Экстренная помощь'**, **'Внутренние болезни'**, **'Лечение ран'**]  
doctor\_elem = [**'Иванов И.И.'**, **'Петров П.П.'**]  
*# ...*@app.route(**'/login/'**, methods=[**'post'**, **'get'**])  
**def** login():  
 **if** request.method == **'POST'**:  
 user = request.form.get(**'username'**)  
 email = request.form.get(**'email'**)  
 phone\_number = request.form.get(**'phone\_number'**)  
  
 direction = request.form.get(**'direction'**)  
  
 direction = direction\_elem[int(direction[-1]) - 1]  
 *# запрос к данным формы* doctor = request.form.get(**'doctor'**)  
  
 doctor = doctor\_elem[int(doctor[-1]) - 1]  
  
 time1 = request.form.get(**'time1'**)  
 time2 = request.form.get(**'time2'**)  
  
 data1 = request.form.get(**'calendar1'**)  
 data2 = request.form.get(**'calendar2'**)  
 **'''  
 user = 'kkk'  
 email = 'aa@aa.ru'  
 phone\_number = '+7'  
 direction = 'Офтальмолог'  
 doctor = 'Иванов И.И.'  
 data1 = '2020-02-12'  
 data2 = '2020-02-15'  
 time1 = '10:00'  
 time2 = '17:00'   
 '''** print(user, email, phone\_number, direction, doctor, data1, data2, time1, time2)  
 Thread(target=take\_zapis, args=(user, email, phone\_number, direction, doctor, data1, data2, time1, time2)).start()  
  
 **return** render\_template(**'login.html'**)  
  
  
*# ...***if** \_\_name\_\_ == **"\_\_main\_\_"**:  
 app.run(host=**'0.0.0.0'**, port=4567)

Основная программа:

**from** selenium **import** webdriver  
**from** selenium.webdriver.support.ui **import** WebDriverWait  
**from** selenium.webdriver.support **import** expected\_conditions **as** EC  
**from** selenium.webdriver.common.by **import** By  
**from** selenium.webdriver.chrome.options **import** Options  
**from** selenium.webdriver.common.keys **import** Keys  
**from** selenium.webdriver.common.action\_chains **import** ActionChains  
**from** webdriver\_manager.chrome **import** ChromeDriverManager  
**import** time  
**import** traceback  
**from** email\_send **import** send\_massege  
**import** datetime  
WINDOW\_SIZE = **"1920,1080"**chrome\_options = Options()  
chrome\_options.add\_argument(**"--headless"**)  
chrome\_options.add\_argument(**"--window-size=%s"** % WINDOW\_SIZE)  
driver = webdriver.Chrome(ChromeDriverManager().install(), chrome\_options=chrome\_options)  
*# driver = webdriver.Chrome(ChromeDriverManager().install())*driver.implicitly\_wait(20)  
  
  
*# ----------------------------------------------------------------------------------------------------------***def** specialty\_selection(a):  
 *# print(driver.find\_elements\_by\_tag\_name("h3"))* way\_elem = WebDriverWait(driver, 30).until(EC.visibility\_of\_element\_located((  
 By.XPATH, **f'\*//h3[contains(text(), "{**a**}")]'**)))  
 way\_elem.click()  
 driver.switch\_to.frame(driver.find\_element\_by\_xpath(**'.//iframe'**))  
 *# print(driver.find\_elements\_by\_tag\_name("button"))* time.sleep(2)  
 tmp = driver.find\_element\_by\_tag\_name(**"button"**)  
 tmp.click()  
 *# driver.switch\_to.default\_content()  
  
  
# ----------------------------------------------------------------------------------------------------------***def** doctor\_choice(doctor\_name):  
 *# driver.switch\_to.frame(driver.find\_element\_by\_xpath('.//iframe'))* dropdown\_menu = WebDriverWait(driver, 30).until(EC.visibility\_of\_element\_located((  
 By.XPATH, **'.//span[@data-hook="dropdown-select-label"]'** )))  
 dropdown\_menu.click()  
 doctor\_elem = WebDriverWait(driver, 30).until(EC.visibility\_of\_element\_located((  
 By.XPATH, **f'.//li[contains(text(), "{**doctor\_name**}")]'** )))  
 doctor\_elem.click()  
  
 *#options = webdriver.ChromeOptions()  
 #options.add\_argument('--headless')  
  
  
# ----------------------------------------------------------------------------------------------------------***def** time\_choice(time\_start, time\_finish):  
 **try**:  
 elem = driver.find\_elements\_by\_xpath(**'.//div//span[@class="ng-binding" and not(@data-hook)]'**)  
 **except**:  
 **return** 0  
 print(elem)  
 **if** len(elem) > 0:  
 **for** i **in** elem:  
 **if** time\_start <= i.text <= time\_finish:  
 i.click()  
 **return** i  
 **return** 0  
 **else**:  
 **return** 0  
  
  
*# ----------------------------------------------------------------------------------------------------------***def** data\_choice(data\_start, data\_finish, time\_start, time\_finish):  
 month = str(datetime.date.today())  
 elem = [month]  
 time = 0  
 **while** elem[-1] <= data\_finish:  
  
  
 elem = driver.find\_elements\_by\_xpath(**'.//td//td[@data-date]'**)  
 *# print(elem[0].get\_attribute('data-date'))* **for** i **in** range(len(elem)):  
 elem[i] = elem[i].get\_attribute(**'data-date'**)  
 elem = sorted(list(set(elem)))  
  
 **for** time\_i **in** elem:  
 print(time\_i)  
 **if** data\_start <= time\_i <= data\_finish:  
 dropdown\_menu = WebDriverWait(driver, 30).until(EC.visibility\_of\_element\_located((  
 By.XPATH,  
 **f'\*//td[@data-date="{**time\_i**}"]'** )))  
 dropdown\_menu.click()  
  
 time = time\_choice(time\_start, time\_finish)  
 **if** time:  
 **break** target\_elem = WebDriverWait(driver, 30).until(EC.visibility\_of\_element\_located((  
 By.XPATH, **f'.//div[contains(text(), "Показать месяц")]'** )))  
 target\_elem.click()  
 **else**:  
 driver.find\_element\_by\_xpath(**'\*//button [@aria-label = "next month"]'**).click()  
  
 **if** time:  
 **break  
 return** i, time.text  
  
  
*# ----------------------------------------------------------------------------------------------------------***def** go\_to\_registration():  
 dropdown\_menu = WebDriverWait(driver, 30).until(EC.visibility\_of\_element\_located((  
 By.XPATH,  
 **'/html/body/main/div/bks-app/div/div[1]/boost-calendar-page/div[2]/boost-visitor-sidebar/section/div[1]/div/button/boost-next-step-label/span'** )))  
 dropdown\_menu.click()  
  
  
*# ----------------------------------------------------------------------------------------------------------***def** reception\_registration(user, email, phone\_number):  
 elem\_2 = driver.find\_elements\_by\_tag\_name(**'input'**)  
  
 elem\_2[0].send\_keys(user)  
 elem\_2[1].send\_keys(email)  
 elem\_2[2].send\_keys(phone\_number)  
  
 button = driver.find\_element\_by\_xpath(**'.//div[@class="sidebar"]//span[@data-hook="next-step-label"]'**)  
 button.click()  
  
  
*# ----------------------------------------------------------------------------------------------------------***def** take\_zapis(user, email, phone\_number, direction, doctor, data\_start, data\_finish, time\_start, time\_finish):  
 **while True**:  
 **try**:  
 driver.get(**'https://zhenyanovokshanov1.wixsite.com/klinika/blank-5'**)  
 specialty\_selection(direction)  
 doctor\_choice(doctor)  
 a, b = data\_choice(data\_start, data\_finish, time\_start, time\_finish)  
  
 go\_to\_registration()  
 reception\_registration(user, email, phone\_number)  
  
  
 **except** Exception **as** e:  
 print(**'Ошибка:\n'**, traceback.format\_exc())  
 print(**'Время занято'**)  
 **else**:  
 print(**"OK"**)  
 send\_massege(email, b, a, user, doctor)  
 **break** time.sleep(10)

Программа отправки сообщений:

**import** smtplib  
**from** email.mime.multipart **import** MIMEMultipart   
**from** email.mime.text **import** MIMEText   
**from** email.mime.image **import** MIMEImage  
**def** send\_massege(addr\_to, time, day, user, doctor):  
 addr\_from = **"\*\*\*\*\*\*\*\*"** password = **"\*\*\*\*\*\*\*\*"** msg = MIMEMultipart()   
 msg[**'From'**] = addr\_from   
 msg[**'To'**] = addr\_to   
 msg[**'Subject'**] = **'Запись на прием'** body = **f"Уважаемый, {**user**}! Вы записаны на прием к {**doctor**} на {**time**} {**day**}"** msg.attach(MIMEText(body, **'plain'**))   
  
  
 smtpObj = smtplib.SMTP\_SSL(**'smtp.mail.ru'**, 465)  
 smtpObj.login(addr\_from, password)   
 smtpObj.send\_message(msg)

Почта и пароль были скрыты по понятным причинам!