

Техническое задание на добавление регистрации и входа пользователя в аккаунт Wix Member Area сайта personalytics.me

1. Общие сведения о проекте

1.1. Краткое описание

Personalytics - разрабатываемая система, которая предназначена для получения статистики фитнес-трекеров пользователей, обработки и анализа полученных данных, представления результатов. Сайт <https://www.personalytics.me> - ресурс, на котором пользователи могут посмотреть информацию о продукте и зарегистрироваться.

1.2. Общие сведения

Тип: сайт

Адрес: <https://www.personalytics.me>

Платформа: Wix

Технологии: JavaScript (Wix Corvid), html

1.3. Цель и задачи

Цель: дать возможность пользователю зарегистрироваться на сайте и входить в свой аккаунт, используя сторонний web-сервис на сервере заказчика. Сервис использует свою реализацию Google OAuth аутентификации.

Задачи:

1. Добавить страницы с необходимыми элементами управления для регистрации, логина пользователя, а также выхода из аккаунта.
2. Для кнопок регистрации и логина реализовать отправку запросов к web-сервису с последующей обработкой ответов.
3. Исходя из ответов, осуществлять или не осуществлять вход пользователя в аккаунт и отображать определенную страницу.

2. Требования

2.1. Требования к web страницам и элементам управления

- 2.1.1. Необходимо добавить новые формы (wix custom lightbox) для окон регистрации и логина.
- 2.1.2. Внешний вид и положение элементов управления новых форм должны соответствовать стандартным Member Signup Default Form, но без полей и кнопки регистрации/входа по логину и паролю и без кнопки регистрации/входа Facebook.
- 2.1.3. Окно, которое отображает Google consent screen, должно открываться в отдельном окне, но не на отдельной вкладке, а на той же, на которой отображается форма регистрации/логина (поверх этой формы, также как при ее отображении в случае использования Member Signup Default Form -> Google login).
- 2.1.4. Кнопка регистрации/логина должна быть html элементом типа <button>.
- 2.1.5. В окне регистрации должна быть ссылка на окно логина и наоборот.

Отображение окон с формами см. в *приложении 1*.

Описание поведения пользовательского интерфейса см. в *приложении 2*.

2.2. Требования к обработке ответов запроса к web-сервису

- 2.2.1. При нажатии на кнопку регистрации или входа должен отправляться GET запрос на тестовый сервер заказчика по адресу <https://core-service-test.herokuapp.com/auth/google>, на котором производится Google OAuth2 аутентификация пользователя с отображением Google consent screen.

- 2.2.2. Если осуществляется регистрация, то при ответе от сервера, говорящем об успешной аутентификации (status 200 OK), необходимо зарегистрировать аккаунт пользователя в Member Area и отобразить страницу по адресу <https://www.personalytics.me/our-prices>
- 2.2.3. Если осуществляется логин, то при ответе от сервера, говорящем об успешной аутентификации (status 200 OK), необходимо осуществить вход пользователя в его аккаунт в Member Area и отобразить страницу по адресу <https://www.personalytics.me/account/my-account> с данными профиля пользователя.
- 2.2.4. При ответе от сервера, говорящем об ошибке аутентификации (status 401 Unauthorized), нужно отобразить страницу по адресу <https://www.personalytics.me/auth-fail>
- 2.2.5. При успешной регистрации в теле ответа будет находиться JSON с данными пользователя: email, first name, last name, которые необходимо сохранить для аккаунта пользователя в Members Collection, чтобы их можно было позже видеть в контактах в Site Member List.

Описание поведения пользовательского интерфейса см. в *приложении 2*.

Описание данных для тела ответа от web-сервиса при успешной аутентификации см. в *приложении 3*.

2.3. Нефункциональные требования

- 2.3.1. Должна быть возможность легко поменять URL для запроса к сервису аутентификации (поменять с тестового на действующий).
- 2.3.2. Необходимо максимально (насколько это возможно) использовать встроенный в Wix механизм автоматического добавления Members Area (Member Page, Member Signup) и аутентификации.
- 2.3.3. Добавление вручную новых окон и элементов управления должно быть сведено к минимуму.
- 2.3.4. Количество добавляемого HTML и JavaScript кода должно быть сведено к минимуму, насколько это возможно.

3. Состав и содержание работ

- 3.1. Изучение технического задания, его согласование и утверждение.
- 3.2. Выяснение деталей, нюансов задачи и уточнения по возникшим вопросам в процессе разработки.
- 3.3. Разработка окон и элементов управления.
- 3.4. Реализация отправки запросов сервису и обработки ответов.
- 3.5. Исправление недоработок и ошибок по результатам тестирования разработчиком.
- 3.6. Исправление недоработок и ошибок по результатам тестирования заказчиком.

4. Требования к тестированию

1. Тестирование разработчиком

Для тестирования необходим аккаунт в Google. При тестировании необходимо предоставить доступ на чтение некоторых данных Google Fit.

Необходимо проводить мануальное тестирование после внесения любых изменений на опубликованном сайте.

2. Тестирование заказчиком

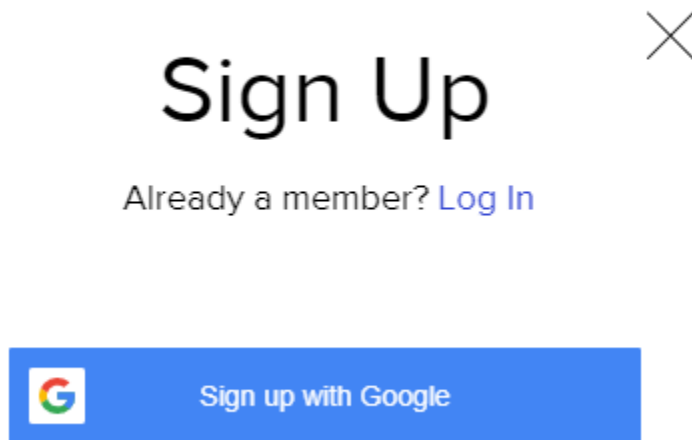
Заказчик тестирует после завершения работы разработчика, в том числе после исправлений ошибок и доработок.

5. Порядок контроля и приемки результатов

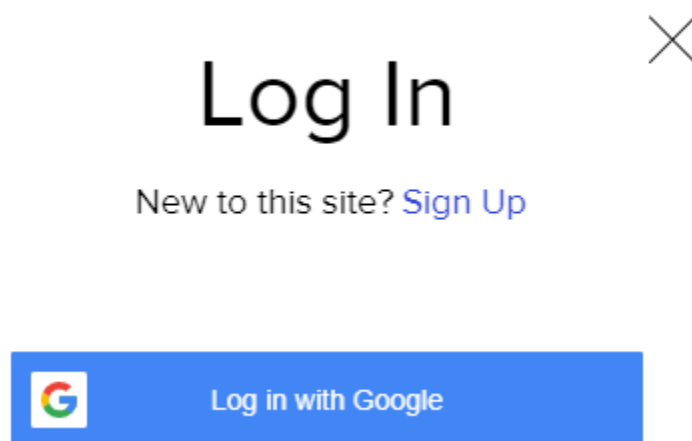
После окончания разработки заказчик просматривает результаты на опубликованном сайте. Производится тестирование. Если после тестирования, будут найдены ошибки или результаты будут не соответствовать каким-либо требованиям, разработчик делает исправления и доработки. Такая последовательность действий выполняется до тех пор, пока заказчик не будет удовлетворен результатом.

Приложение 1. Отображение окон с формами

Окно регистрации:



Окно входа:



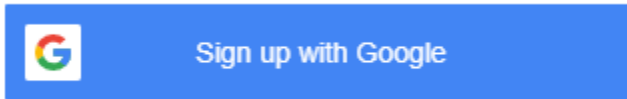
Приложение 2. Описание поведения пользовательского интерфейса

Для регистрации:

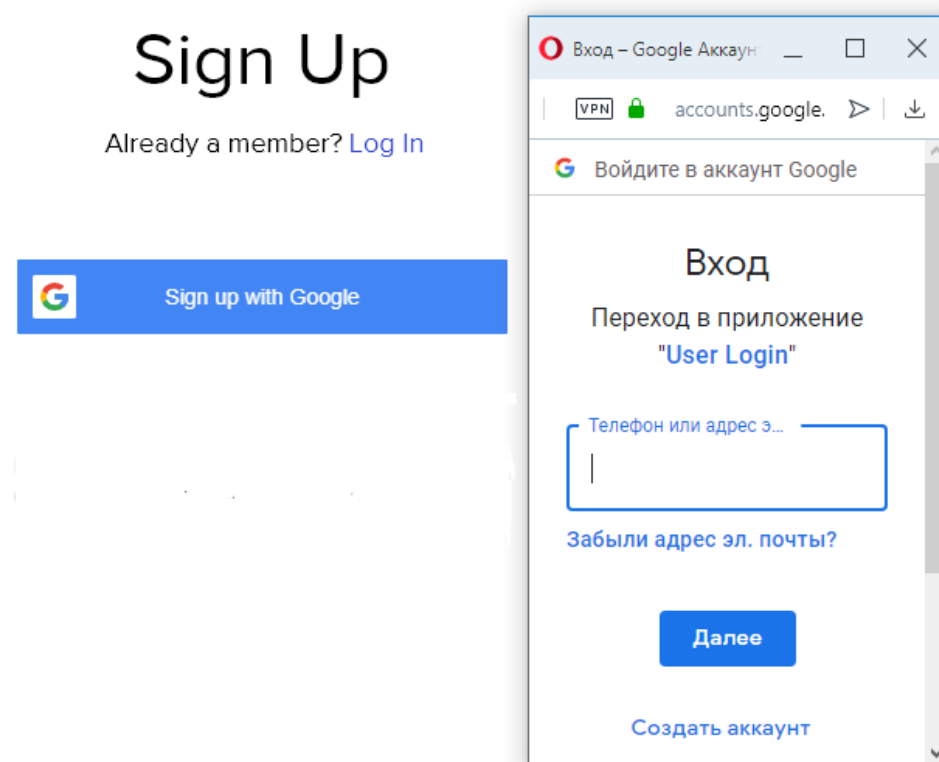
1. Пользователь нажимает:



2. Открывается окно регистрации.
3. Если пользователь хочет зарегистрироваться, он нажимает:



4. Ему отображается окно Google consent screen, вызываемое web-сервисом:



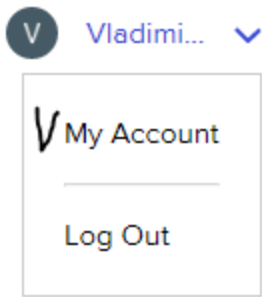
5. Он выполняет вход в Google account или регистрируется в Google, если это необходимо. А также дает разрешение на доступ к данным.
6. Ему отображается страница сайта:



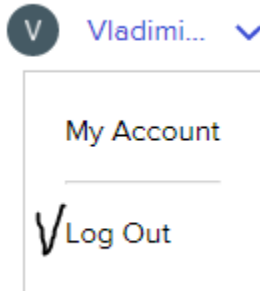
[how it works](#) [our story](#) [get started](#)

Choose your best option

7. Для перехода к настройкам аккаунта пользователь нажимает My Account:



8. Для выхода из аккаунта пользователь нажимает Log Out:



Для входа в аккаунт:

1. Пользователь нажимает:

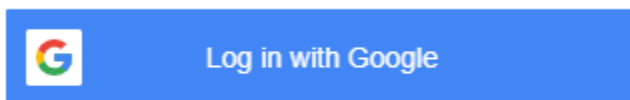


2. Открывается окно регистрации.

3. Если пользователь хочет перейти ко входу в аккаунт, он нажимает на ссылку 'Log in':

Already a member? [Log In](#)

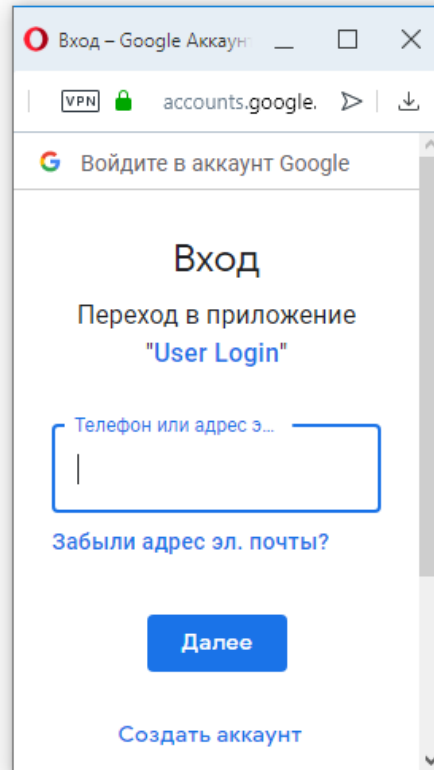
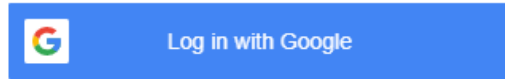
4. Если пользователь хочет осуществить вход, он нажимает:



5. Ему отображается окно Google consent screen, вызываемое web-сервисом:

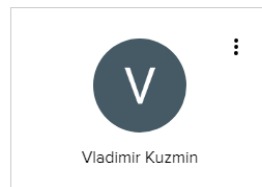
Log In

New to this site? [Sign Up](#)



6. Он выполняет вход в Google account, если это необходимо.
7. Ему отображается страница сайта с настройками аккаунта:

 Vladimi... ▾



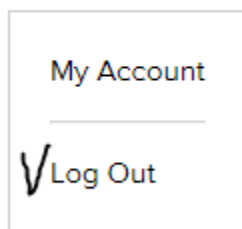
My Account

View and edit your personal info below.

Login Email:

8. Для выхода из аккаунта пользователь нажимает Log Out:

 Vladimi... ▾



Приложение 3. JSON-схема для данных в ответе от web-сервиса при успешной аутентификации

Схема:

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "type": "object",
  "title": "User's data",
  "properties": {
    "email": {
      "type": "string",
      "title": "User's email",
      "examples": [
        "email@email.com"
      ]
    },
    "firstName": {
      "type": "string",
      "title": "User's first name",
      "examples": [
        "Alexander"
      ]
    },
    "lastName": {
      "type": "string",
      "title": "User's last name",
      "examples": [
        "Dmitriev"
      ]
    }
  },
  "required": [
    "email",
    "firstName",
    "lastName"
  ]
}
```

Пример JSON, оформленного по схеме:

```
{
  "email": "email@email.com",
  "firstName": "Alexander",
  "lastName": "Dmitriev"
}
```