

GymHeroes

**A directional and motivational
workout guide**

Liam Bradley, Ryan Nemec, Ken Tanoue

GymHeroes:

GymHero allows you to be able to access instructional information regarding your personal health, While experiencing your favorite motivational music **ALL IN ONE PLACE.**

The Motivation Behind Our Motivator:

People want all of their functionalities in one place.

Many apps, like Youtube for example, stop playing on your mobile device when you shift away from the app.

With GymHeroes, you are able to have your critical instructional material and your streaming music all in one place without interruptions.

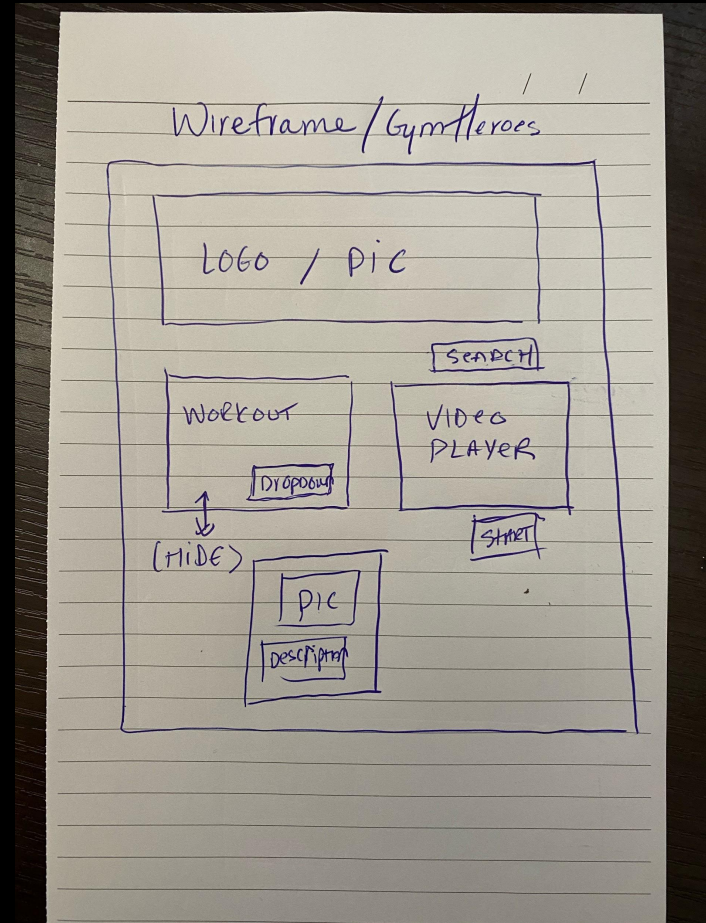
Who You Are:

A person who needs a little motivation to get going.

Someone who has sheltered-in-place and wants that “personal trainer” experience.

An avid “gym-goer” who wants to listen to their favorite streaming youtube music without interruptions.

Wireframe Image



GYM HEROES

SEARCH FOR MUSIC GENRE



WORKOUTS

Please select what muscle group you would like to workout:

SELECT WORKOUT



START

Task Leader Breakdown:

Liam Bradley - CSS Framework

Ken Tanoue - Wreg.de API.

Ryan Nemec - Youtube API.

API Research

YouTube:

Initially, we were going to use a traditional music player like Spotify, or Last.FM. These sources required OAuth2 certification and access keys.

Youtube allowed us access to streaming music with a simple api key, as well as having a huge library of music to assist us in our vision.

Wreg.de

This was one of the few free exercise apis that we found that would display images of the specific exercises which were segregated by category.

This resulted in creating multiple ajax calls to get the data required from multiple endpoints.

API by default was in German.

Query URL Breakdown:

YouTube:

An AJAX call was made to the Youtube API with the search functionality being the endpoint to grab on to.

Youtube allows you to have access to almost any aspect of their UI wso narrowing it down to 1 video that was embeddable was necessary.

“Music” is an already included keyword in the search function so that it displays playlist based off things other than just genres and artists.

```
function getVideo() {  
  var youtubeForm = $("#autocomplete-input").val()  
  var musicGenerator = ("music%20" + youtubeForm);  
  $.ajax({  
    type: 'GET',  
    url: 'https://www.googleapis.com/youtube/v3/search',  
    data: {  
      key: "AIzaSyAJ1ag4z7gAcPM3dQ14tX7C0YqKiYeK6B4",  
      q: musicGenerator,  
      part: 'snippet',  
      maxResults: 1,  
      type: 'video',  
      videoEmbeddable: true,  
    },  
    success: function(data){  
      $("#youtube").attr('src', 'https://www.youtube.com/embed/' + data.items[0].id.videoId + "?a  
      console.log(youtubeForm)  
    },  
  })  
}
```

Query URL Breakdown:

Wreg.de:

Multiple ajax calls were required to retrieve the data from this API.

1st ajax call was used for the “category & description of each exercise”.

2nd ajax call was used for the specific exercise images.

API key was required to be entered as a header for validation to work.

Default German language was able to be translated within the url by inserting “language=2” for English.

```
//exercises on api are listed by id.
//each category is linked to individual apis, which resulted in using
function getExercises(id) {
  var filteredresults = [];
  $.ajax({
    type: "Get",
    //filter by language by adding language query to url. English is (2) in url link.
    //Exercise category includes exercise description.
    url: "https://wger.de/api/v2/exercise/?format=json&language=2&category=" + id,
    dataType: "json",
    headers: {
      Authorization: "Token 16a85a599865174319c1a5f12bcd324e58d7507",
    },
    success: function (resultscategory) {
      getExercisesuccess(resultscategory, filteredresults);

      $.ajax({
        type: "Get",
        url: "https://wger.de/api/v2/exerciseimage/" + id + "/thumbnails/?language=2",
        dataType: "json",
        headers: {
          Authorization: "Token 16a85a599865174319c1a5f12bcd324e58d7507",
        },
        success: function (resultsimgs) {
```

Code Highlights

An Event Listener was added to link all of our exercise JS functions to a drop down menu on our home page.

Based on the users input, our application will generate results based on specific exercise category. (Abs, chest, back)

```
//event listener for drop choice options linking workout id with thumbnails and descriptions for each.
var choiceid = 8
$(document).on("click", ".dropChoice", function (event) {
    event.preventDefault();
    choiceid = $(this).attr("id");
    console.log(choiceid);
})
document.addEventListener('DOMContentLoaded', function() {
    var elems = document.querySelectorAll('.dropdown-trigger');
    var instances = M.Dropdown.init(elems, options);
    var workoutType = $(".dropChoice").val()
    console.log(workoutType)
});
```

WORKOUTS

Please select what muscle group you would like to workout:

SELECT WORKOUT

SEARCH

MUSIC GENRE

ABS

LEGS

ARMS

SHOULDERS

BACK

CHEST

CALVES

W

Pl

group you would like to workout:

New Library : *Garlic.js*

Garlic.js allows you to save the text written in a form before it has been submitted automatically adding it to the LocalStorage.

All you have to do is add: `data-persist="garlic" method="POST"`

To your input field in the HTML.

Demo



Future Development:

What what you do with more time?

YouTube:

Refine the search ability.

Potentially utilize more music focused API's like Spotify.

Have customer playlists depending on search variables.

Wreg.de

Filter out results better to get more specific data needed for our application.

Include more features such as a nutritional plan and schedule to compliment the exercises that the user selects.

Questions

