# Season Flash

# **Challenge 5**



#### Challenge 5: Sportify - Season Summary

WHY: Engage with athletes and empower him to engage with his fans using his season activity data.



#### Notes from Pre-event:

Also introduce some kind of content about their life

General info
And specific info regarding the athlete
Start with the basic starting information
Tool that is used. A new tool and get a real benefit
Enhancing the data with photos and pictures
Allibaba has a tool for data visualisation
Processing is a javalike programming language. That is used by animator to create graphics
and real time animations.
Olig that create experience with it
Didg that create experience with it
Didg that create experience with it
Didg that of the the processing is a proper to the programming that any the programming that any the programming that any the programming that any the programming that the processing the processing that the processing the processing the processing that the processing that the processin





#### Brainstorming Data Aggregations



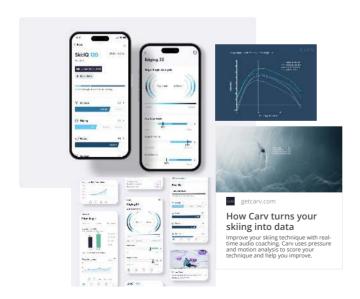


## Using the Stages of Team Development Team effectiveness is enhanced by a

Team effectiveness is enhanced by a team's commitment to reflection and on-going evaluation. In addition to evaluating accomplishments in terms of meeting specific goals, for teams to be high-performing it is essential for them to understand their devel...





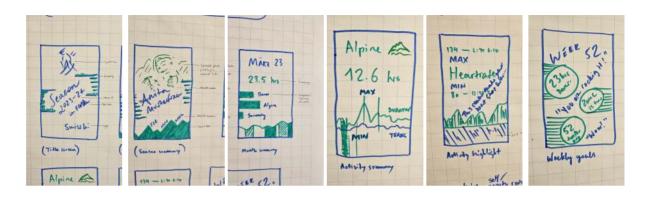








### Sketch



## Storyboard









52 bpmhr
The road ahead is clear: a disciplined focus on reps will pave the way to gold!

	Call to action Athlete: we want your data!	
Who		
Where		
What		

	Title screens Summary of the season, sport, athlete
Who	
Where	
What	

Who	
	Where
	where

Maxir	Activity highlights na, dates-times, trends, recommendations
Who	
Where	
What	

	Summary Info-bubbles with top stats	
Who		
Where		
What		

#### Desktop (node-based, point & click)



#### Motion graphics software | Adobe After Effects

With Adobe After Effects, the industrystandard motion graphics software, you can take any idea and make it move. Design for film, TV, video, and web.



#### @ cables.gl

#### cables

Cables is a tool for creating beautiful interactive content. With an easy to navigate interface and real time visuals, it allows for rapid prototyping and fast adjustments.



#### friction.graphics

#### Friction

Friction is a powerful and versatile motion graphics application that allows you to create vector and raster animations for web and video platforms with ease.



#### pitivi.org

#### Pitivi, a free and open source video editor for Linux

Pitivi is a Free video editor with a beautiful and intuitive user interface, a clean codebase and a fantastic community. Join us!



#### TOOLL3 - Open Source Motion Graphics

TOOLL3 is an open source software to create realtime motion graphics. We are targeting the sweet spot between realtime rendering, graph-based procedural content generation and linear keyframe animation and editing. This combination



Open Source Compositing Software For VFX and Motion Graphics







www.stablevideo.com

## Stable Video | Generate Videos with Al

Create videos with text or image, turning your concepts into captivating cinematics.

## Helpful SVG generators for patterns & textures



#### A Collection of Free Grunge, Vintage & Concrete Textures

Genuine high-res textures from aged walls and grounds. Authentic grunge, vintage, and concrete overlays to give your photos and images that extra grit.



#### ooorganize: SVG background grid pattern generator

An online SVG generator for various kinds of grid patterns that can then be used as visual decoration in web

## Data processing



The Jupyter Notebook is a web-based interactive computing platform. The notebook combines live code, equations, narrative text, visualizations, interactive dashboards and other media.





## Dataset provided (anonymized)





Heart Rate Zones:

Zone 1: 80-144 bpm

Zone 2: 145-159 Zone 3: 160-169

zone 3: 160-169

Zone4: 170-184

Zone 5: 185-105

## Open data sources

Name	URL / API	Example
FIS (all olympic athletes data)	Timing & Data  Fis globally governs sking and snowboarding and oversees over 7,000 country. Ski Jumping, Nordic Combined, Freestyle, Snowboard, and more.	Audi
Wikidata (olympic results, general info)	Wikidata.org Wikidata:REST API The Wikibase REST API is an OpenAPI- based interface that allows users to interact with, retrieve and edit Items and statements on Wikibase instances - including of course Wikidata. For more information about REST, see the Wikipedia entry on representat	Wikidata:WikiProject Sport results  Welcare to the WikiProj Sport results  Welcare to the WikiProject Sport result The WikiProject Sport result The WikiProject to dedicated to the crastion and uplespe of Bernat WikiProject sport results will help of describe spoils writted in metadata in a uniform way across all spo-
BFS (Swiss open data)	opendata.swiss  Daten  The opendata swiss portal is a joint project of the Confederation, cantons, communes and other organizations with open government data available to the general public in a central catalogue. opendata.swiss is operated	opendata.swiss  Mobilité de loisirs - Ski de fond - opendata.swiss Réseau homologué des pistes de ski de fond
Whoop (sleeping patterns)	developer.whoop.com  WHOOP API Docs    WHOOP for Developers  HTTP API Docs for WHOOP	Training vs sleep in professional sport Data analysis of sleep and training performance of a professional athlete.
Ski 2DPose (computer vision)	Www.epfl.ch  Ski 2DPose Dataset  Overview We created a new 2D pose dataset for aligne shing that can be used for further research connecting computer vision and sports sciences. While there are many large-scale human pose datasets, most don't feature many images of skiers and usually	Papers with Code - Ski- Papers with Code - Ski- Pose PTZ-Camera Dataset This studiese part of bitches career alpha stars per froming gall staffer appear from a caltraster, their projected 20 per from a caltraster, their startese

```
Jupyter Sports Hackdays Code Last Checkpoint: 8 hours ago (autosaved)
                                                                                                   zones
                                                                                                                           1/
                                              Widgets
                                                                                                                       Trusted
        Edit
               View
                      Insert
                              Cell
                                     Kernel
                                                            ~ [222]
    In [258]: | import pandas as pd
                   # Ensure 'start date' is in datetime format
                   xc training data['start date'] = pd.to datetime(xc training data['start date'])
                   # Extract the month and year from 'start date'
                   xc training data['month'] = xc training data['start date'].dt.to period('M') # 'M' stands for month
                   # Group by the month and sum the total training seconds for each month
                   monthly summary = xc training data.groupby('month')['total seconds'].sum()
                   # Convert seconds to hours and minutes
                   def convert_seconds_to_hm(seconds):
                       hours = seconds // 3600
                       minutes = (seconds % 3600) // 60
                       return f"{int(hours)}h {int(minutes)}m"
                   # Apply the conversion function to the monthly summary
                   monthly summary hm = monthly summary.apply(convert seconds to hm)
                   monthly summary hm dict = {str(month): time for month, time in monthly summary hm.items()}
                   # Display the monthly summary without seconds
                   print(monthly summary hm)
                   month
                   2023-05
                                81h 0m
                   2023-06
                              101h 25m
```

115h 35m

103h 0m

2023-07

## Compositing

## Processing (code, web-based)





## Export to video



p5.js | vple
After my first few days getting into creative coding / compform, I reached a point where I wanted to share some animations I created. The problem was that I didn't know how to get an animated gif. The sketches on p5.js (editor) are all rendered in an H...

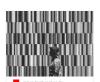


Creating videos from Processing's p5.js with CCapture.js and ffmpeg

A tutorial on creating videos from Processing's javascript library p5.js with the help of CCapture.js and ffmpeg.

## Motion graphics





courses Archive \* tim rodenbröker creative coding
One of the most exciting and maybe even unsettling discoveries in the learning process of Creative Coding in Graphic Design [...]

pypi.org

Creative coding in Python

## Data visuals



Quick guide into creative coding with p5.js
A beginner friendly introduction to generative art

ditor,p5js.org

DwD: Using chart.js
with p5.js by
jeffThompson -p5.js
Web Editor

A web editor for p5is, a jaw.6ript

A web editor for p

#### Advanced animation





anime.js

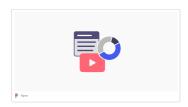
Javascript animation engi



Using React with p5.js (React Hooks and Context)

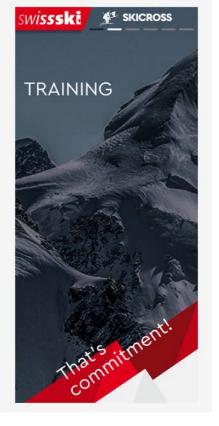
Using React with p5.js This project demonstrates how to combine React (including the latest features such as hooks and context) and p5.js: Multiple p5 sketches on a same screen; Multiple instances of one sketch mounted in a same component; Mount sketche...

## Wireframes









## **Work in Progress**

```
♥0 □0 90 th 📵 SAVEASFOR
                                                                                                                                                                      □ ±
       mybata.iotatifainingiime,
         myData.MostCommonSportsType,
         myData.longest workout info.duration
30 }
33 timer = 0:
34 rect_height = 50;
35 rect margin = 5;
+ 37 function draw() {
       timer++;
      if (timer > 1200) { return }
       background(100);
       image(myBG, 0, 0, WINDOW_WIDTH, WINDOW_HEIGHT);
       fill(50):
       rect(0, WINDOW_HEIGHT-logo_h-20, WINDOW_WIDTH, logo_h+20);
       c = color(255);
       c.setAlpha(150);
      textFont(myFontBold, 30);
      for (i = 0; i < 3; i++) {
        xx = 3.2 * timer / (i + 1);
        xx = min(xx, (1 + i) * 100);
        yy = rect_margin + (i * rect_height);
         fill(c):
        rect(0, yy, xx, rect_height - rect_margin);
         fill(255);
        text(info[i], xx + 5, yy + rect_height / 1.8);
      alpha(1.0);
       fill(255);
      textAlign(CENTER);
       textFont(myFontBold, 70);
       text('SEASON\nFLASH', CENTER_X, CENTER_Y + min(100, timer));
```

## Sonification / Soundtrack



## Mozilla Web Speech API





