Technological Feasibility Analysis
Version 2
11/3/2021
Rehab Remote

Project Sponsor
Dr. Zachary F. Lerner
BiOMOTUM, Inc.

Faculty Mentor
Felicity H. Escarzaga

Team Members
Kylie Cook
Brandon Roberts
Robert Bednarek
Katarina Marsteller

Table of Contents

Introduction

In the United States, 1 in 345 children has cerebral palsy, a set of neuromuscular disorders effective around birth. Cerebral palsy can reduce the mobility of those that have it. Around 58.9% of children with cerebral palsy can walk independently, 7.8% walk with assistance and 33.3% have limited or no walking ability [3].

The current treatments for cerebral palsy consist of medication, therapy, and surgery. Biomotum stands within the therapy category by developing a battery-powered ankle exoskeleton that increases walking speed by 32%, stride length by 21%, and improves efficiency by 29% [2]. The business currently works with children with cerebral palsy. However, they wish to expand to people with other conditions that reduce their mobility, such as strokes, surgeries, broken bones, and more.

Biomotum's current workflow starts with the use of the ankle exoskeleton. First, they place a child on a treadmill with the exoskeleton and other physical assistance. Next, the researchers measure how much energy the child uses during testing and track the child's movements. The test is then repeated with different forms of physical assistance while they continue to collect data. After a session of the exoskeleton's use, the patient will access their data through the Biomotum phone application. The patient can visualize their steps, tork, duration of use, and more within the application.

The issue within the business workflow is a lack of desktop visualization. Biomotum currently extracts the raw data by unhashing the file and running the data through code on MATLAB. By the end, the researchers visualize the data through MATLAB, a series of steps that the average patient could have difficulty understanding.

To create a solution for Biomotum's problem, Team Rehab Remote plans on making a web-based portal that visualizes the data collected from the ankle exoskeleton. Features include:

- Charts that show steps, tork, duration of use, and exoskeleton activity
- Pings Amazon Web Services (AWS) database once per hour
- Interactive dashboard with a visualizer tool and summary statistics
- A patient directory that asks for their name and DOB than goes to -> sessions -> files of sessions
- Global statistics of trials per hospital, person, number of devices active, etc. for admin
- Downloadable charts and data

Technological Challenges

Technological challenges discuss the hurdles discovered for this project. The challenges are briefly described and can be utilized as an overview of the challenge. The challenges' relevance to our overarching problem that we plan to solve with our visualization web application will be described. Finally, these challenges will be further analyzed in the following section to determine the best solution.

1. Extracting and Organizing Data

The data is only currently understandable by Biomotum professionals who are experienced with the MATLAB language and code. First, Python code translates the data into a CSV(Comma-Separated Values) file, limiting the number of visualization tools to translate the raw data appropriately. Then, MATLAB code extracts the data and creates graphs. Our client's data extraction process needs to be compared by others to accurately and effectively be used in the project.

2. Account Authentication/Creation

Setting up an account creation system where user data is easily accessible and very secure raises careful planning and considerations. Our client, Dr. Zachary F. Lerner, has asked us to ensure that those with a device or a device manager can make accounts only through a sign up link. It was also specified that a specific user should only be able to see their account data versus seeing all account user data.

3. Data Visualization Platform

The organized data should be accurately translated into a graphical image. This graphical image should show progress over time, for example, throughout a session or sessions for a single patient or clinic. It should also show stand-alone statistics, such as the number of steps taken in total with the exoskeletons, for a single patient or clinic. Finally, the data must be displayed to the user in an understandable manner that is constantly updated and effectively shows a session's progress.

4. Downloadable Raw Data

The user may desire the raw data in a downloadable format. Therefore the data must be presentable and easily transferable to other tools. The data itself would

be provided via a CSV file for easy manipulation and reuse. In addition, the charts should be provided in a PDF(Portable Document Format) to be able to download and understand the charts. The PDF would be formatted much like a standard report. It would contain a graphical representation, and some of the stand-alone statistics of the session reported.

Technical Analysis

In order to be on the path of project success, the major technical issues of extracting and organizing data, visualizing the data, securing the website, and including the ability to download the data need to be analyzed. Therefore, each issue has been studied with multiple alternatives to choose from to get the job done. Below, the alternatives are rated, and one is chosen among all problems.

1. Extracting and Organizing Data

The ability to extract and organize the data from Biomotum is considered a major technical issue due to possible incompatibility within languages used to further analyze the data obtained once the extraction/organization is complete.

Biomotum's Process

Biomotum keeps the data extracted from their ankle exoskeleton in AWS (Amazon Web Services), a cloud platform. They use a Python script that utilizes boto3 API and a secret hash to pull information from the AWS database. The secret hash is used to avoid non-authorized users from accessing and possibly editing that information. Once the data downloads from AWS, the company uses a MATLAB script to extract and organize the data from the CSV file created by the Python script.

Choosing a Process

In order to decide on what extraction process will be used, they must satisfy the following requirements:

- **Prior development knowledge:**
  The key to the success of our project. Learning a new language takes more time than future maintainers might have. This development time would be better spent developing and maintaining the project rather than learning a new language.

- **Compatibility with web hosting service:**
  With our project being a web-portal, it is key that our chosen data science language can proficiently connect with many chosen services to avoid unnecessary delay in our development cycle.

- **Ability To Access Data:**
  In order for a process to be chosen, it has to do the minimal work of having the ability to access CSV data in the first place. This can sometimes be done by using a data science language by itself, although some languages need extra libraries to access files.

Alternatives

Below are the programming language options for extracting and organizing the data.

- **MATLAB:**
  Since Biomotum uses MATLAB to extract and organize the data from given CSV files, the obvious option is to try the same process. MATLAB was created in the late 1970s by Cleve Moler. Average users of MATLAB come from engineering, science, and economic backgrounds and normally use the language for numerical computing [5].

- **Python:**
  Python is a possible alternative in the project to connect the front-end and back-end of the application. The language was taught in CS 126 (Computer Science I) at NAU. It was created in 1991 by Guido van Rossum [6].

Analysis

Below is the exploration of the desired characteristics of each possible web platform alternative. Each alternative will start with a score of 0 out of 3 requirements that they need to cover and a point will be added when the alternative covers a requirement.

- **MATLAB:**
  - **Prior development knowledge:**
    As Biomotum currently uses MATLAB to extract data from multiple CSV files and create graphs using that data, MATLAB gets a +1 to its comparison scores. Once Biomotum takes over the website, they will not need to learn any new language and can reuse code that they already have written.

- **Compatibility with web hosting service:**
  Since Biomotum already uses Wix as their current website hosting service, then the web application will also be created through Wix. Unfortunately, MATLAB is not compatible with Wix and does not receive a point for comparison scores.

- **Ability to access data:**
  Since Biomotum uses the language to access CSV data already, the language gets a +1 in comparison scores. Although, as an example, a small CSV file was created with random data along with a small MATLAB script to access it, and MATLAB was able to read from the CSV file as well as organize the data (Figure 1.1).

```
opts = detectImportOptions('Testing\test1.csv');
preview('Testing\test1.csv',opts)
```

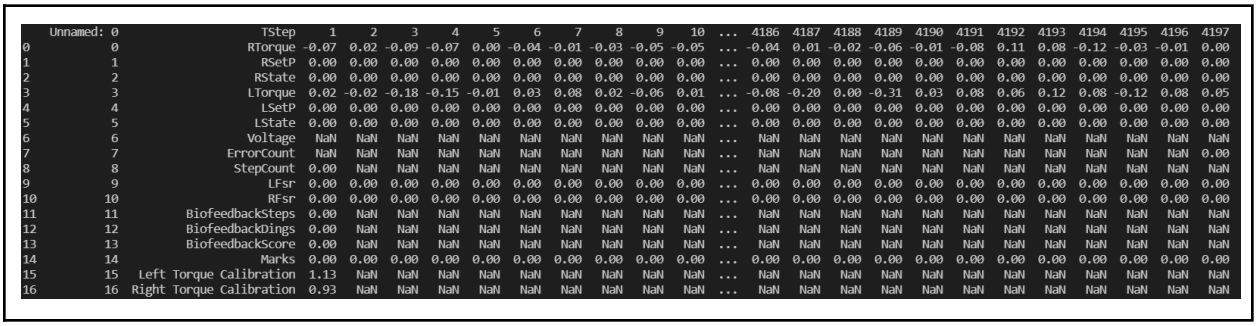
Var1	Var2	Var3	Var4	Var5	Var6	Var7
{'Steps' }	100	NaN	NaN	NaN	NaN	NaN
{'Torque'}	1	2	1	2	3	1
{'Miles' }	2	1	3	5	6	8

Figure 1.1: MATLAB Data Access

- **Python:**
  - **Prior development knowledge:**
    Since Biomotum also uses Python to create the CSV files from the AWS database, Python can be considered as prior development knowledge, giving it a +1 in comparison points.

  - **Compatibility with web hosting service:**
    As mentioned above, Wix will be used as the web hosting service, and Python is also not compatible. Python is compatible with Google Charts, which can create a website to embed within a Wix site. This earns Python a +1, since MATLAB is not compatible with Google Charts.

- **Ability to access data:**
  A necessity to the language that is used when accessing the data is accessing the entirety of a CSV file given from Biomotum. A Python script was created to read from and organize data from an example CSV file from Biomotum, and successfully did its part, giving it a +1 (Figure 1.2).



	Unnamed: 0	TStep	1	2	3	4	5	6	7	8	9	10	...	4186	4187	4188	4189	4190	4191	4192	4193	4194	4195	4196	4197
0	0	RTorque	-0.07	0.02	-0.09	-0.07	0.00	-0.04	-0.01	-0.03	-0.05	-0.05	...	-0.04	0.01	-0.02	-0.06	-0.01	-0.08	0.11	0.08	-0.12	-0.03	-0.01	0.00
1	1	RSetP	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	2	RState	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	3	LTorque	0.02	-0.02	-0.18	-0.15	-0.01	0.03	0.08	0.02	-0.06	0.01	...	-0.08	-0.20	0.00	-0.31	0.03	0.08	0.06	0.12	0.08	-0.12	0.08	0.05
4	4	LSetP	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	5	LState	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	6	Voltage	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7	7	ErrorCount	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.00
8	8	StepCount	0.00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
9	9	LFsr	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	10	RFsr	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
11	11	BiofeedbackSteps	0.00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
12	12	BiofeedbackDings	0.00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
13	13	BiofeedbackScore	0.00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
14	14	Marks	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15	15	Left Torque Calibration	1.13	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
16	16	Right Torque Calibration	0.93	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Figure 1.2: Python Data Access

Chosen Approach

The options that were analyzed are rated out of 3, which comes from each requirement fulfilled by each option (Table 1).

Table 1: Extract and Organize Data Scores

Language	Rating
MATLAB	2/3
Python	3/3

Since Python fulfills all 3 requirements, it will be the chosen language to extract and organize the data from CSV files extracted from AWS.

Proving Feasibility

In order to make sure that Python will work in our project, we will need to access the Biomotum AWS cloud directly, download recent CSV files, and run them through the Python code to generate any file needed for the next step. For the stretch goals, we will also need to continuously sync data from the cloud, run the files through the script, and have files organized every couple of minutes.

2. Account Authentication/Creation

Setting up an account creation system where user data is easily accessible and also very secure raises a need for careful planning and considerations. Our client Dr. Zachary F. Lerner has asked us to ensure that those only with a device or are a device manager are able to make accounts. It was also specified that a specific user should only be able to see their own account data versus being able to see all account user data.

- **Desired Characteristics**
  Selecting a set up method where only specific users can create an account and then can only see their desired statistics is vital to the success of our project and client confidentiality.

  - **Account creation:**
    The ground floor of our application account system. The ability to create a user account is our main concern. The platform that we choose must support this feature with options for flexibility.

  - **Account security:**
    Federal law requires that we take all possible steps to prevent the disclosure of sensitive patient health information without patient consent at all costs. Because of this it is vital that our account creation system is secure and only displays data relevant to the account holder.

  - **Scalability:**
    The system we create has the possibility to be ever growing. The system that we put in place will need the ability to scale to handle the possibility of hundreds of users creating and accessing their accounts.

- **Alternatives**

  - **Wix**

    - **Account creation:**
      Wix, as shown in Table 2 scored a 5 out of 5 for Account Creation. This is for a variety of reasons such as account creation is fast and simple as that is Wix's intention. The account management system is superb as administrators can manage accounts in a variety of ways such as

limiting restrictions to certain pages and so on. The cherry on the top of the cake is the ability to turn on the account setup feature is as easy as clicking a button.

- **Account security:**
  Security, as shown in Table 2, has scored a 3 for a few simple reasons. These are primarily because of the lack of information that we have on the security measures in place for Wix. While we can do the basic needed security steps such as username and password and account restrictions, we are unable to verify the backend is secure. [9]

- **Scalability:**
  Scalability as shown in Table 2 has scored a 3 for a few simple reasons. These are due to the way that Wix account management is set up. Each time an account is set up the account administrator is in charge of the account details. This could lead to future problems for the administrator when the account pool grows large enough. [10]

- **Custom Setup**

  - **Account creation:**
    Account Creation, as shown in Table 2, has scored a 2. This is because of the extensive work that would be required to set this up and to expand features in the Account Creation regard. Each feature would have to be programmed and any features that come after would have to be manually implemented as well.

  - **Account security:**
    Security scores a 5 as shown in Table 2. This is because we would have total control over the system we create. We would be aware of all potential flaws and fix them accordingly.

  - **Scalability:**
    Scalability as shown scores a 4 in Table 2. This is due to the ability to set up the account system in such a way where administrators would not be required to monitor and set up each account. The reason this does not score a 5 is any features that we do not implement currently would need to be created later.

- **WordPress**

- **Account creation:**
  WordPress as shown scores a 3 in Table 2. This is due to the need to use plug-ins. Should we decide or have the need to change to another system (plug in) in the future, that would be a huge hindrance to the task. [1]

- **Account security:**
  Security as shown in Table 2 scores a 3. This is because of the unreliability of plug-ins. Each plug in would have its own security system in place with its own flaws that would need to be learned and accounted for.

- **Scalability:**
  Scalability as shown in Table 2 scores a 3. This is due to the need to use different plug-ins again that are yet again unreliable and each coming with their own flaws. Should we change our plug in at any time, it would be a reworking of the core of the system.

Comparison/Chosen Approach

The criteria used above will be used to rate each alternative language on a scale of 1-5 to make a decision for the best approach. 1 being the worst and 5 being the best.

Table 2: Account Authentication/Creation Scores

	Creation	**Security**	**Scalability**
Wix	5	3	3
Custom Setup	2	5	4
WordPress	3	3	3

Proving Feasibility

As indicated by our rubric, Wix and WordPress are our primary options. WordPress seems to be the lesser of the two options as the unfamiliarity of the platform and the unreliability of having to interact with the multiple plug-ins we would have to incorporate. Wix has been our chosen platform for creating accounts and ensuring account security. The ability to have administrators

enable accounts and the way that account settings are flexible and customized per user is essential for success.

3. Data Visualization Platform

Choosing a platform where we can host and visualize various graphical representations of the data being pulled in from the AWS database is the most important part of this project. Once the data is organized, it will need to be effectively translated into diagrams that make the data very easy to understand. It will need to display specific data related to a patient as well as global statistics related to certain clinics for further analysis. The platform should be easy to use and have good support for graphing capabilities.

- ## Desired Characteristics

  Selecting a platform that has a balanced mix of graphical representation tools flexibility and scalability is essential to this project's success. The scoring is based on a 1 to 5 scale with 1 being the worst and 5 being the best. The following metrics will be used to evaluate each platform:

  - **Visual Quality:**
    The website must be visually appealing while maintaining the overall functionality of the page. For example, the user would want to see a clean design in line with Biomotum’s current website and mobile app.

  - **Graphical support:**
    Creating and maintaining graphs that are updated in real time can be very complicated and hard to implement if you do not have a lot of experience with it. For example, a user does not want to see data that was posted weeks ago because it becomes irrelevant to the current progress being collected by the exoskeleton. In addition, the user must receive not only an up-to-date visual, but an appealing report pertaining to the graphics. Having support for creating visually appealing reports and graphical representations of the data being collected and managed will dramatically reduce the development time of the project.

  - **Scalability:**
    This website has the potential to be expanded to include many other datasets for different types of patients or even different exoskeleton technologies. As previously stated, the platform must be suitable for not only children, but adults and professionals as well. For example, the

platform must be able to expand to include datasets that may be hospital or clinic specific, perhaps even nationwide. The platform must be scalable so that it can easily be expanded in the future to support other features and functionalities.

- ## Candidates

  - **Wix:**
    Wix is the web development platform that we were encouraged to use by Biomotum. It is a free website builder that uses drag and drop tools as well as the ability to create custom JavaScript code for specific functionality. In addition, Wix provides many frameworks and features, such as website templates and hosting services which are free to any user who uses their platform.

  - **Ruby on Rails:**
    Ruby on Rails is an open-source application framework written in the Ruby programming language. Ruby on Rails has been used to develop platforms such as Twitch, a live streaming platform, and GitHub. Ruby on Rails focuses on programmer's use rather than promoting a single solution. This platform is highly flexible and a very popular framework to build web applications in.

  - **Bootstrap:**
    Bootstrap is a free open source CSS framework for front end web development. This platform focuses on integration and use of other technologies to achieve a powerful and responsive web application. It can be easily installed and has convenient documentation that explains the tools and features available in Bootstrap.

- ## Analysis
  Below is the exploration of the desired characteristics of each possible web platform alternative:

  - **Wix**

    - **Visual Quality:**
      Wix, as shown in Table 3, scored a 5 of 5 for Visual Quality. Wix has built-in website templates which cater to the design the user desires.

Upon first creating the website, the user is provided with design keywords such as “Modern” or “Vintage”. These keywords when selected assist the system in selecting text fonts, layouts, and overall presentation of the website. In addition, the templates are also customizable in order to further refine the website.

- **Graphical Support:**
  Wix, as shown in Table 3, scored a 4 of 5 for Graphical Support. Wix has tools which aid in the visualization of data. These visualization tools are able to present graphs as the client requested. However, the tools themselves are purchased only through the Wix market and have very little documentation on said tools.

- **Scalability:**
  Wix, according to Table 3, scored a 4 of 5 for Scalability. Wix allows for multiple sites to be linked together and has various tools to do so. However, the number of sites that can be made is limited and has advertisements posted for the free access Wix provides. In addition, bandwidth issues tend to occur with complex sites similar to what will be constructed. Therefore, the scalability could be challenging however there are tools which could help to mitigate the

Table 3 : Wix Analysis Scores

Metric	**Visual Quality**	**Graphical Support**	**Scalability**
Score (out of 5)	5	4	4

- **Ruby On Rails**

  - **Visual Quality:**
    Ruby, according to Table 4, scored a 3 for visual quality. There are plenty of products which Ruby was utilized such as Twitch and GitHub. The designs are very distinct and easy to understand documentation for customizing the site entirely. However, these sites are quite simple with little to none of the features we would like to implement. In

addition, Ruby has a tendency to have a slow boot speed that will end up limiting how much the site may be able to hold and display.

- **Graphical Support:**
  Ruby, according to Table 4, scored a 3 out of 5 for graphical support. Ruby has extensive libraries and extensive documentation for implementing active record management functions. However, the documentation does not specifically state anything in regards to graphical representation.

- **Scalability:**
  Ruby, according to Table 4, scored a 4 out of 5 for scalability. While many platforms that use Ruby are large scale platforms, the product is geared toward individuals and therefore may only be scaled to include more patient data for display. Regardless, Ruby ensures that the ability to scale requires minimal adjustment in terms of code.

Table 4 : Ruby On Rails Analysis Scores

Metric	**Visual Quality**	**Graphical Support**	**Scalability**
Score (out of 5)	3	3	4

- **Bootstrap**

  - **Visual Quality:**
    Bootstrap, according to Table 5, scored a 3 of 5 in visual quality. Bootstrap has a plethora of options for customization. All these tools make for very complex displays. However, the features may not all appear across all platforms and some features may be unresponsive.

  - **Graphical Support:**
    Bootstrap, according to Table 5, scored a 2 of 5 for graphical support. Bootstrap allows for images to be displayed in a web page. However, our product requires an "active graph", meaning that as data comes in, the graph is updated. Bootstrap would require other outside support in order to achieve this.

  - **Scalability:**

Bootstrap, according to Table 5, scored a 3 of 5 for scalability. Bootstrap has the capability for large scale CSS and Javascript and is very integratable with most other technologies. Unfortunately, Bootstrap is very HTML and CSS dependent which limits what technologies we can use for our active graphs.

Table 5 : Bootstrap Analysis Scores

Metric	**Visual Quality**	**Graphical Support**	**Scalability**
Score (out of 5)	3	2	3

- **Comparison/Chosen Approach**

Table 6: Overall Possible Candidate Scores

	Visual Quality	**Graphical Support**	**Scalability**
Wix	5	4	4
Ruby on Rails	3	3	4
Bootstrap	3	2	3

After scoring the three web platforms, we chose to use Wix as our web platform to host our web application. Although Ruby on Rails might have been a better long term solution in terms of scalability, it would be a big learning curve for our team and it might lead to us not accomplishing the minimum viable product for the project if we run into issues. For this reason we will use Wix as a simple and efficient all in one solution to host a web app with beautiful graphs and visuals.

- **Proving Feasibility**

Wix is a successful web building platform that we should have no issue getting set up very quickly. This will allow us to create beautiful graphical representations of the data from AWS so that doctors and researchers can easily interpret the data. The biggest challenge for us will be refreshing ourselves with JavaScript so that we can create the backend connection to the AWS database. Going forward, we will do further research to learn the

best way to do this and experiment with different methods to get the best result.

4. Downloadable Raw Data

The data that will be pulled in from AWS will be used to create and update charts of various information from the exoskeleton in use but we also need to be able to download the raw form of this data. Since the original data is stored in CSV files, we will want to extract the data back out into a downloadable CSV file. The Web Portal will have statistics and charts for each specific patient as well as global statistics so there needs to be a “download raw data” link associated to each of these hierarchies of data groups. Also, we will need a “downloadable statistics” link for each of these hierarchies that allows a PDF of the statistics and charts shown to be downloaded for easy printability.

- ## Desired Characteristics

  Selecting the most efficient and flexible programming language that allows us to create downloadable files which can be integrated into an online platform has multiple sought-after characteristics. The following criteria that will be used to evaluate each language are as follows:

  - **File Support:**
    The chosen language must have support for different file types, more specifically PDF and CSV files.

  - **Ease of implementation:**
    It is important that the language is easy to implement into the rest of the software. Since we will be using Wix, the language must be compatible with Wix and be somewhat easy to implement.

  - **Scalability:**
    It is anticipated that the software created will be added to and adapted in the future for other uses. This means that we should use a language that scales easily and can be manipulated to expand the use case of downloadable data.

- ## Alternatives

  - **Python:**
    Python is an interpreted language high-level general-purpose programming language. It is very easy to use and has a multitude of libraries and support from other platforms because of how popular it is in almost all use cases. This is the language that Biomotum created a script for pulling in data from their AWS database which will be helpful to get a head start on development.

  - **JavaScript:**
    JavaScript is a text-based programming language used in both client and server side to make interactive web pages. This language is used primarily by web developers to create web applications that deliver an optimal user experience. For those reasons, this language will have to be implemented in the interaction with a web portal for the best user experience.

- ## Analysis

  Below is the exploration of each possible alternative language along with a score for each metric.

  - **Python**

    - **File Support:**
      Python does have support for downloading files through its extensive libraries and supported modules. Specifically there is a pandas package that allows for the conversion of simple text files into CSV files [4]. In terms of file support, Python earns a score of 3/3 due to having straightforward support for creating CSV files.

    - **Ease of Implementation:**
      Python does have multiple ways to download files. These methods include “urllib.request.urlretrieve”, “requests.get + manual save”, and “wget.download” [1]. However Wix does not support the Python language but it is compatible with Google charts which could be used to get data into the Wix platform as mentioned in section 1 of the Technological Analysis. Python receives a score of 1/3 for the ease of implementation because although it’s possible, it requires an unnecessarily complicated work-around.

- **Scalability:**
  Python is a very scalable language due to its vast number of supported libraries and modules. It has multiple different solutions to almost any problem. This allows for the application backend to be easily scaled up to include more functionality. Python receives a score of 3/3 because it is a modular object oriented language that has lots of documentation on using various packages.

- **JavaScript**

  - **File Support:**
    JavaScript has support for creating and downloading a CSV file. As explained in javatpoint.com, “you can collect the data from the HTML page in a CSV file and download it easily” [7]. JavaScript receives a score of 3/3 because it has full support for CSV files with potential issues.

  - **Ease of Implementation:**
    Javascript has the easiest implementation by far because it is the only programming language that Wix supports. It is important to use a language that can be integrated easily into the backend of our web application which will be developed with Wix. JavaScript receives a score of 3/3 because it has full support and integration with Wix so that will make implementing downloadable links into the web app much easier with less potential for issues.

  - **Scalability:**
    JavaScript is one of the most scalable languages out there, especially in terms of web development. Although it might take a bit extra time to get it set up than Python, the long term web scalability of JavaScript far outweighs any possible cons. JavaScript receives a score of 3/3 for scalability because it is the best language to use for developing web applications in the computer industry.

Comparison/Chosen Approach

The criteria used above will be used to rate each alternative language on a scale of 1-3 to make a decision for the best approach. 1 being the worst and 3 being the best.

Table 5: Downloadable Data Metric Scores

	File Support	**Ease of Implementation**	**Scalability**
Python	3	1	3
JavaScript	3	3	3

After scoring the two chosen alternative languages, it is clear that JavaScript is the winner for downloading raw data. It is a very flexible language that is easy to use and can be integrated onto the Wix website very easily.

- Feasibility

  It has been proven that JavaScript will be the easiest language to implement a way to download CSV and PDF files of the data in a Wix web application. Although the current team does not have heavy experience with JavaScript, it is the best language for downloading CSV or PDF files with the use of Wix and it will be the best for web scalability as well. Also, there is thorough documentation that covers this area of downloadable data which will greatly aid the current team as well as any future developers that take on this project in the long term.

Technology Integration

The current implementation plan would consist of 4 main components: Extracting the dataset, the data sorting, the implementation of the data into graphs inside of the web portal, and the creation of updated downloadable data links.

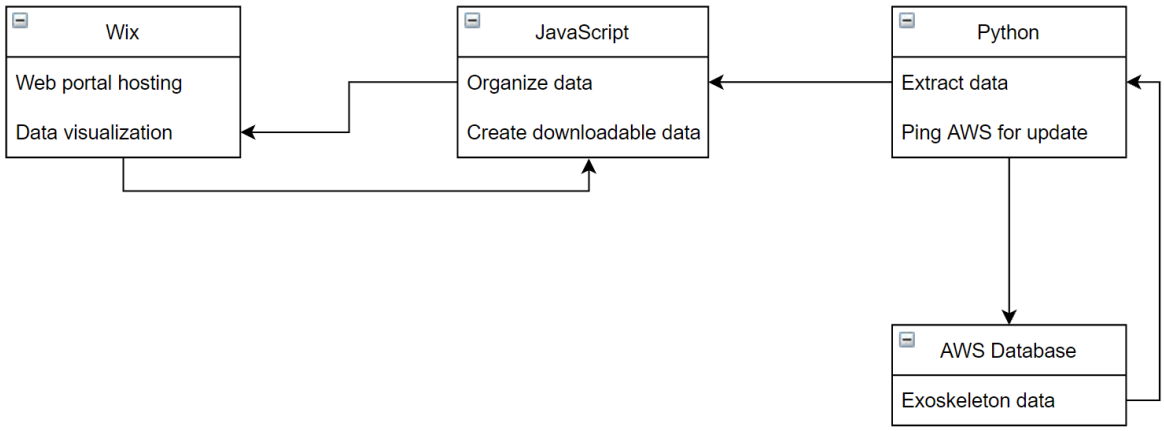


Figure 2 : Proposed Architecture Solution
Python pings the AWS database to extract updated data which is used by JavaScript to organize data, create downloadable data, and allow Wix to visualize the data.

1. Extracting the Data

Extracting the data is the ground floor of our project. It is a vital part of the project in the sense that if extracting the data method breaks or is flawed it will hinder the efficiency of the final result. The current implementation is extracting the data from the AWS server using Python. Python is a simple, robust, reliable language that can easily implement our script.

2. The Dataset

The dataset has many aspects and how we go about retrieving and organizing our data is essential to the success of our project and the future development of this project. Organizing and retrieving our data will be done with Python and JavaScript. Python will be primarily responsible for retrieving the data from the AWS web server while the job of JavaScript will be to do any necessary sorting and organizing of the data for easier use in the web-interface.

3. Web Interface

The web interface is in charge of a visually appealing interface for potentially countless users. The web interface also has many other responsibilities such as interacting with large amounts of data and needs to remain responsive since this is where the graphical representation of the data will occur. For these reasons we aim to use WIX as our source to solve these problems and make use of the simple interface and graphical templates that are provided.

4. Downloadable Data

Lastly, we will want to have links that download the raw data that is being used to create the graphs shown in the web portal. The portal will also contain a link to download a PDF version of the graphs themselves for printability and saving for future reference. As discussed in the Technical Analysis section 4 we found that JavaScript would be the best language to create these downloadable files

Conclusion

While there are different options for cerebral palsy rehabilitation, Biomotum has created an incredible high-tech system that will track the progress of a patient and eventually assist others in need of rehabilitation. Our project will assist patients and researchers involved with Biomotum by helping to understand the data that the ankle-exoskeleton collects.

We plan first to figure out the process of taking and manipulating the exoskeleton data. Our challenges within the topic consisted of how we extract the data from the given CSV files, organize the extracted data, what technology we will use to visualize the data, and how we will provide the ability to download the data. Our options ended with Python to extract, JavaScript to organize, Wix to visualize data, and JavaScript to allow the integrated download ability.

After taking time to research each of our options in every topic provided, we are confident to proceed with our decisions and move to the next steps in the project. We will be able to provide examples of these working technologies within our technology demo, where we hope to start making an impact within Biomotum.

References

[1] West, Z. (2021, January 3). *3 simple ways to download files with python*. Medium. Retrieved November 5, 2021, from https://betterprogramming.pub/3-simple-ways-to-download-files-with-Python-569cb91acae6.

[2] *Science & Outcomes*. Biomotum. (n.d.). Retrieved November 5, 2021, from https://www.biomotum.com/science.

[3] Centers for Disease Control and Prevention. (2020, December 31). *Data and statistics for Cerebral Palsy*. Centers for Disease Control and Prevention. Retrieved November 5, 2021, from https://www.cdc.gov/ncbddd/cp/data.html.

[4] *Data to fish*. Data to Fish. (n.d.). Retrieved November 5, 2021, from https://datatofish.com/convert-text-file-to-csv-using-Python-tool-included/.

[5] Wikimedia Foundation. (2021, November 4). *Matlab*. Wikipedia. Retrieved November 5, 2021, from https://en.wikipedia.org/wiki/MATLAB.

[6] Wikimedia Foundation. (2021, November 5). *Python (programming language)*. Wikipedia. Retrieved November 5, 2021, from https://en.wikipedia.org/wiki/Python_(programming_language).

[7] *Not found: JavaScript-Create-and-download-CSV-file tutorials - javatpoint*. www.javatpoint.com. (n.d.). Retrieved November 5, 2021, from https://www.javatpoint.com/JavaScript-create-and-download-csv-file.

[8] *What is the best platform to develop web applications? (2021)*. Thinklogic. (n.d.). Retrieved November 5, 2021, from https://www.thinklogic.com/post/what-is-the-best-platform-to-develop-web-applications-2021.

[9] Kohai, S. (2018, March 11). *[solved] filter page data by logged-in user*. Velo by Wix. Retrieved November 5, 2021, from https://www.wix.com/velo/forum/tips-tutorials-examples/filter-page-data-by-logged-in-user.

[10]V, S. (2018, April 3). *User specific content*. Velo by Wix. Retrieved November 5, 2021, from https://www.wix.com/velo/forum/coding-with-velo/user-specific-content.

[11] *How to create a user database for a members-only website*. dummies. (n.d.). Retrieved November 5, 2021, from https://www.dummies.com/programming/web-services/how-to-create-a-user-database-for-a-members-only-website/.