

GNG1103  
**Design Project User and Product Manual**

**BeEco:**  
**The Recycling Sorting System**

Submitted by:

BeEco Friendly INC. C2

Griffin Worboy, 300119873

Usama Shahzad, 300111792

Cem Kiyik, 300073910

Nader Mrad, 300118507

Kristen Janzen, 300107082

18/04/21

University of Ottawa

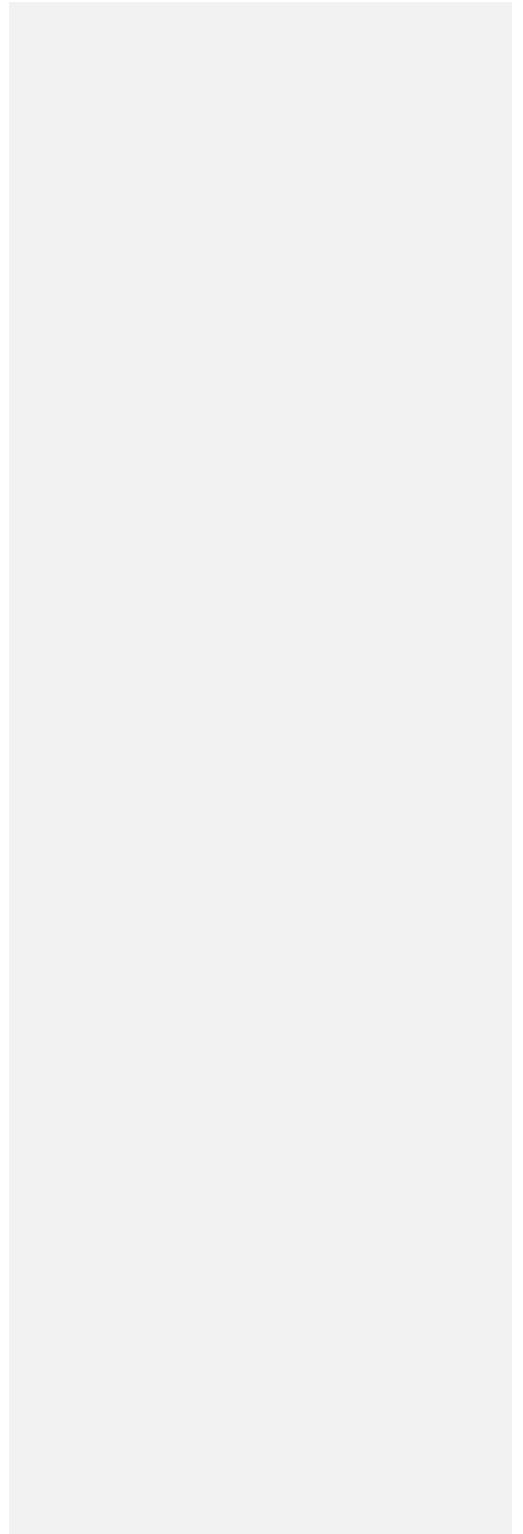
## Table of Contents

Table of Contents	ii
List of Figures	v
List of Tables	vi
List of Acronyms	vi
1 Introduction	1
2 Overview	1
2.1 Cautions & Warnings	4
3 Getting started	4
3.1 Set-up Considerations	4
3.2 Accessing the System	4
3.3 System Organization & Navigation	4
3.4 Exiting the System	4
4 Using the System	5
4.1 BeEco Website	5
4.1.1 Points	6
4.1.2 Calendar	6
4.1.2 Scan	6
4.1.2 IBM Watson	6
5 Troubleshooting & Support	7

5.1	Error Messages or Behaviors	7
5.3	Maintenance	7
5.4	Support	7
6	Product Documentation	7
6.1	Prototype 1	8
6.1.1	Building the Code	9
6.1.1.1	Python	9
6.1.1.2	HTML	12
6.1.1.3	CSS Code	19
6.1.1.4	Java Script	34
6.2	Prototype 2	42
6.3	Prototype 3	42
6.4	BOM (Bill of Materials)	43
6.4.1	Equipment list	43
6.5	Testing & Validation	44
6.5.1	Testing Prototype 1	44
6.5.2	Testing Prototype 2	45
6.5.3	Testing Prototype 3	47
7	Conclusions and Recommendations for Future Work	48
APPENDICES		

APPENDIX I: Design Files

49



## List of Figures

<b>Figure #1:</b> Final Product: BeEco the Universal Recycling Sorter	3
<b>Figure #2:</b> The BeEco User Interface Menu	5
<b>Figure #3:</b> Shows the Desktop Version of the Website	8
<b>Figure #4:</b> Shows the Mobile Version of the Website	8
<b>Figure #5:</b> Necessary Imports	9
<b>Figure #6:</b> Connecting Python to HTML	10
<b>Figure #7:</b> Connection Watson Cloud Server to the Code	11
<b>Figure #8:</b> Decision Mechanism	12
<b>Figure #9:</b> Main Page (HTML)	15
<b>Figure #10:</b> Scan Page (HTML)	16
<b>Figure #11:</b> Calendar Page (HTML)	18
<b>Figure #12:</b> Result Page (HTML)	19
<b>Figure #13:</b> CSS Code to Create BeEco	33
<b>Figure #14:</b> How to use Java Script to use BeEco	42
<b>Figure #15:</b> Results and Accuracies of Testing Recyclable Materials Using the IBM Watson Visual Recognition	47

## List of Tables

<b>Table #1:</b> Acronyms	vi
<b>Table #2:</b> Needs and Their Importance	2
<b>Table #3:</b> Bill of Materials (BOM)	43
<b>Table #4:</b> Equipment List and Details on the Requirements or Use	43
<b>Table #5:</b> Beta Testing of the First Prototype: Testing the User-friendliness Metric	44
<b>Table #6:</b> Beta Testing of the Second Prototype: Testing the Identification of Recyclable Material (Client Need)	45
<b>Table #7:</b> Results of Alpha Testing Prototype Three, Using IBM Watson and BeEco	46
<b>Table #8:</b> Referenced Documents	49

## List of Acronyms and Glossary

Table 1. Acronyms

Acronym	Definition
JS	Java Script
CSS	Cascading Style Sheets
BOM	Bill of Materials
UI	User Interface
AI	Artificial Intelligence

## 1. Introduction

The project of “Waste Management” was a project assigned to the teams by the client Mitch Bouchard. Mitch Bouchard is part of a family business in the field of mechanical part supply. This project has a goal of improving the quality of recyclables and decreasing waste in landfills by creating a system to assist people in sorting their recyclables properly. Though many individuals believe that recycling is enough to make a change in the world this is not the case when looked at on a large scale. Many recyclables are missorted and end up in landfills and/or shipped overseas to countries that have more room for landfills than Canada does. In fact, many people do not know how to recycle and opt for throwing everything into the garbage instead resulting in further pollution in our environment. Creating a product to help people learn and understand how to sort their recyclables and garbage easily could help in the big picture of managing and reducing the amount of waste on the planet.

The product developed was named BeEco, developed by the team/company BeEco Friendly Inc.. The name was created to sound like the “eco”, or “be eco” so that the company’s name would sound like “Be eco-friendly”. The name of the product and the company therefore reflecting the theme of recycling and environment conservation.

The product BeEco, uses IBM Watson Virtual Recognition to determine the material that is being scanned is made of, BeEco can then indicate to the user which recycling bin the object should be placed in.

The User and Product Manual (UPM) will outline the metrics and testing criteria on which the prototypes and testing were developed. The bill of materials and stages of testing and prototyping were also outlined. Then the product development was described including the IBM Watson software and website development. Finally, the stages of using the product were outlined as well as the Results and a discussion of future for the product. This UPM provides all the information necessary for the users to effectively use the BeEco recycling sorting system and for prototype documentation.

## 2. Overview

Not everyone knows how to recycle and worse many people are not recycling properly. Many people throw recyclable material in the waste which fills up landfills or they throw their recycling in the wrong bin (or any bin). This is a surprisingly large issue as this decreases the quality of the materials produced by the recycling plants. When the quality of the materials is decreased, they will not get reused rather thrown away making recycling futile. By using Beco, landfill waste can be reduced and the quality of recycling increased! This cost affective, fun and user-friendly system helps users recycle correctly and efficiently.

The purpose of this app is to help the user identify the right bin to place their recyclable materials if they are unable to identify it. This is done by using the camera feature on their phone to scan the material, which then will inform the user on the bin that they should place the material.

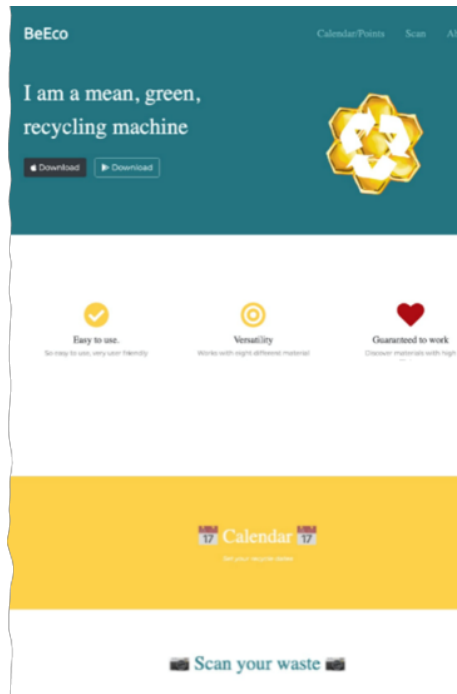
This system was developed based on the needs and wants of the client Mitch Bouchard, who stated that he needed the product to be cost effective, user friendly, versatile and accessible and able to determine recyclable materials as well as identify the correct bin to place them in. These needs and their importance's can be summarized as follows:

**Table 2: Needs and their importance**

Need Number	Need	Importance
1	The product is cost effective.	3
2	The product has a user-friendly interface.	2
3	The product determines the eligibility of items to be recycled.	1
4	The product determines the items respected disposal locations.	1
5	The product is versatile and can be used by many people.	3

When comparing to the other apps that we benchmarked which were Recycle! and Grow Recycling Beco mimics some of the best features from those apps but with its own improvements and innovative systems to go above and beyond. From Grow Recycling came the inspiration for the points system that can be found in BeEco. Grow Recycling provided a fun and educational game to teach and encourage children to recycle. BeEco Friendly Inc. decided that BeEco should act in a similar way by using something to motivate adults to use the product as well as want to recycle. Therefore, the points system was created to incentives and promote recycling by a point system in which the user will get points if they recycle properly. However, this is still to be developed as it requires sponsors. Beco also allows the user to choose a Recycling day, so they can go back and reverence it the next time they recycle so it serves as a reminder to the users to ensure that they have put their Recycling materials outside on the designated day. This was inspired by the Recycle! Application as the main function (and highest rated feature) is reminders so that users never forget their recycling day.





**Figure #1:** Final Product: BeEco the Universal Recycling Sorter

The BeEco website has three main features, the Camera or Scanning feature, the Calendar feature and the Point system.

First is the Scanning or Camera feature. This feature allows the user to scan their material using a camera or webcam on their device and the website will inform them on what bin to place their recyclable material in. This is currently a working feature in the app that works for cardboard, paper, metal, glass, and plastic accurately.

Secondly, BeEco offers the calendar feature. This feature allows the user to add their recycling day, BeEco will then remind them when to place their recycling out for pickup.

The final feature is the Point System. Points are displayed on the points section and inform the user how many points they have. These points are earned by scanning and recycling materials.

## 2.1 Cautions & Warnings

This project has Creative Commons-Attribution- Non-Commercial license. This means that credit must be given to BeEco Friendly INC when using this system as a part of a future project, the project must also be non-commercial in order to use BeEco as a part of this project.

## 3. Getting started

Some materials and installations must be made in order to properly recreate and/or operate the BeEco software. This section outlines the general considerations which should be taken before starting this project. This includes the set-up considerations, how to Access the system, system organization and navigation and exiting the system.

### 3.1 Set-up Considerations

The setup should consist of a powerful laptop or desktop computer with a webcam or a camera, an IBM Watson account and have python, Java script and Cascading Style Sheets ready to use for website creation. Ensuring a stable internet connection and a reliable computer are essential to the proper setup.

### 3.2 Accessing the System

Currently the website is using a computer as the server but if a domain and server were to be bought then it could run on the server and could be accessible to the outside world.

### 3.3 System Organization & Navigation

The BeEco webpage is organized into three sections or tabs, the Scan tab, the points tab and the calendar tab. To navigate the system the titles can be clicked on in order to open the webpage tab.

### 3.4 Exiting the System

To exit the system the user must close window using the “x” (red) to close their browsers and windows or quitting using the menu bar on the computer the system is running on.

Commented [KJ1]: Explain how they would go about creating the domain we did or accessing one we provide them with (client).

Commented [CK2R1]: they cannot reach our thing unless we deploy it

Commented [KJ3R2]:

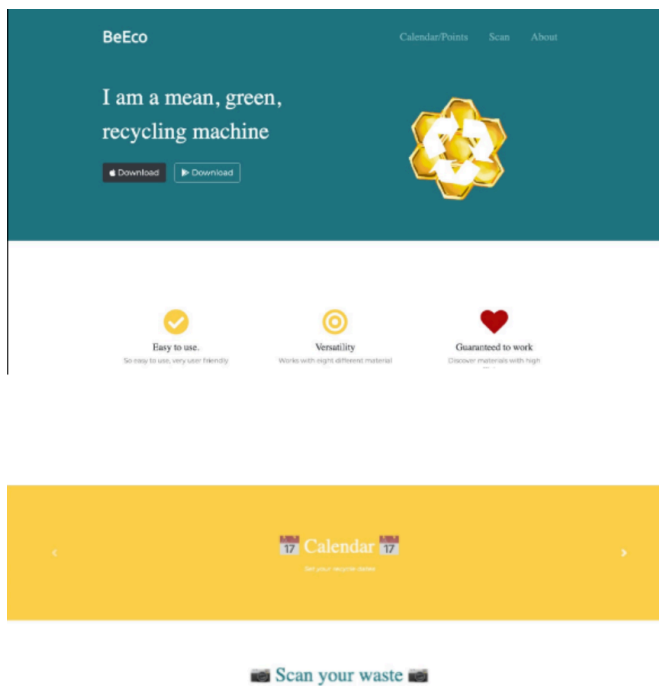
Commented [KJ4R2]:

## 4. Using the System

The following sub-sections provide detailed, step-by-step instructions on how to use the various functions or features of the BeEco Recycling Sorting System.

### 4.1 BeEco Website

The BeEco site is used to recommend the correct recycling bin to place the scanned object in. When arriving on the website the homepage is displayed as follows:



**Figure #2:** The BeEco User Interface/Menu

From the homepage the user has the option to pick from the three options displayed on the main menu bar. The three options consist of scan, points and calendar.

#### **4.1.1 Points**

The “points” tab directs the user to the points page. Here your points will be stored, points are earned when you recycle products. These points can be used towards discounts and other sponsored prizes to promote recycling. This function would ideally be stored in cookies or account information for individuals using the product.

#### **4.1.2 Calendar**

The “calendar” tab can be clicked to view the calendar. To use the calendar, function the user would have to input the dates of their local recycling pickup day. This information would also be stored in the same way as the points information. This tab would then allow for the user to be reminded or check when their recycling pickup is in order to ensure that their recycling goes out on time. These dates would be displayed on the calendar.

#### **4.1.3 Scan**

The scan option opens the main function of the website. The scan function will allow the user to scan the object they wish to recycle, and it will direct them to the correct bin. In order to scan the object, the user will have to hold the item in the camera frame shown, the object should be fully displayed with the least other materials in the photo frame. Ideally the image should be clear, focused and not blurry (decent camera quality) to yield the best results from the object scan. To scan the object the “scan” button is pressed, the allows the website to take a picture of the object in order to run it through the IBM Watson software as described above.

#### **4.1.4 IBM Watson**

In order to expand the database, you need to go to the IBM Watson Visual Recognition website, and you need to create a new model. Once a new model is created you should upload images to it and they should be selected and then labeled. Once all your images are classified you can train the database and now the database is updated.

## 5. Troubleshooting & Support

If an error occurs when operating the BeEco website or the IBM Watson software it must immediately be addressed. If this is a commonly occurring error or an error avoidable by maintenance, it is described below. If this section does not address your concern, please reach out to the contacts provided.

### 5.1 Error Messages or Behaviors

The user may receive the message Key not found: 'collections' error message. This means that the material couldn't be recognized. Secondly, the user may receive the message 'This site cannot be reached'. This message means that server is not running.

### 5.3 Maintenance

To avoid failure of the prototype, continue to train and increase the IBM Watson database. This will ensure that the system has enough images to compare and run properly. Also ensure to keep the host computer up to date on software updates and check that the internet connection is strong enough to run the software. Ensuring that the computer has enough memory and power to run the software continuously and smoothly should also be monitored.

### 5.4 Support

For technical Support please reach out to IBM Watson help services when dealing with errors in the IBM Watson software. For any other issues regarding the BeEco website that cannot be otherwise be resolved Cem Kiyik at [ckiyi015@uottawa.ca](mailto:ckiyi015@uottawa.ca).

## 6. Product Documentation

The final design is a software. It is a Website made using HTML5, CSS, JavaScript, and Python (Flask). Website also uses an API taken from IBM Watson/Visual Recognition. IBM Watson is used to recognize the object. Python is used to control the operation. It renders HTML pages and HTML pages adopts JavaScript and CSS required.

Tensorflow/Keras libraries could also be used to recognize objects but IBM Watson gave a better success rate and therefore it is chosen. Moreover, a unity app/XCode App/ Android Studio App could be made instead of the website but a website was built because a website can

be used from any device that has internet communication and since it has proper CSS it works well in a mobile device as well.

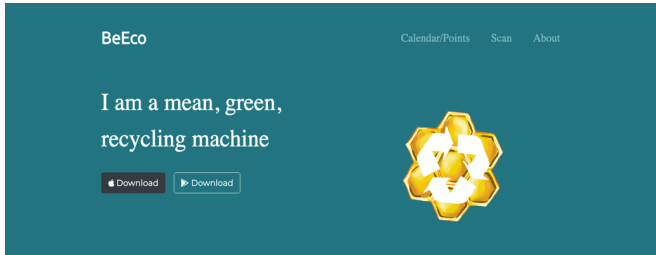


Figure #3: Shows the desktop version of the website

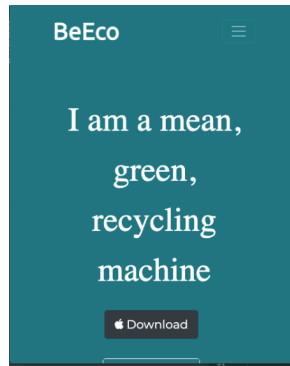


Figure #4: Shows the mobile version of the website

### 6.1 Prototype 1

The main goal for prototype one was to create the main website up or the UI and ensure that it functioned properly and provided a user-friendly experience. There were four main parts to the development of this prototype. The four parts are the homepage, the scan page, the calendar page, and the points page. We used HTML, CSS, and JavaScript to build the website. There are three main directories on the website. The first directory is the scan button. Once the Scan Button is pressed it will bring you to a new page where it accesses your camera. When the calendar button is pressed it will open an

updated calendar that shows the present date. Under the points tab a random number of points are generated due to the limited time and nature of the project. The main system of the scan function was the focus and tweaking the other features was a part of the future development plan.

### 6.1.1 Building the Code

This section outlines the code required to develop the BeEco website. Following the code, a recreating of BeEco can be achieved.

#### 6.1.1.1 Python

This section outlines the code required in python to run the website here the necessary inputs, connecting python to HTML, connection the IBM Watson cloud server to the code and the coding of the bin decision system.

```
from flask import Flask, render_template
import datetime
import cv2
import random
import time

app = Flask(__name__)

#Get_year
now = datetime.datetime.now()
year = now.year

#Get_points
points = random.randint(0,1000)
```

Figure 5: Necessary imports

```
''' Connecting to the html file'''  
  
@app.route('/')  
def home():  
  
    return render_template("index.html", year=_year, points=_points)  
  
@app.route('/calendar')  
def cal():  
  
    return render_template("example.html")  
  
@app.route('/scan')  
def scan():  
  
    return render_template("scan.html")  
  
@app.route('/identify')  
def identify():  
    '''Analyze the image'''  
    camera_port = 0  
  
    ramp_frames = 30  
  
    camera = cv2.VideoCapture(camera_port)  
  
    def get_image():  
        retval, im = camera.read()  
        return im  
  
    for i in range(ramp_frames):  
        temp = get_image()  
    print("capturing image...")  
    camera_capture = get_image()  
    file = "scanned_image.jpg"  
    cv2.imwrite(file, camera_capture)  
  
    del camera  
    print("Image received")
```

Figure 6: Connecting python to HTML



```

from ibm_watson import VisualRecognitionV4
from ibm_watson.visual_recognition_v4 import FileWithMetadata, AnalyzeEnums
from ibm_cloud_sdk_core.authenticators import IAMAuthenticator

apikey = 'IFkd9hrcKE0k2P851YmtL5En8z4XMURU8-N74NeMhQ4'
url = 'https://api.us-south.visual-recognition.watson.cloud.ibm.com/instances/aad5af72-1ae1-4e51-bad8-520cde7c7a76'
collection = '7e818abc-2c89-4d74-b482-deed6413282e'

authenticator = IAMAuthenticator(apikey)
service = VisualRecognitionV4('2018-03-19', authenticator=authenticator)
service.set_service_url(url)

r = random.randint(0, 100)
path = 'scanned_image.jpg'

with open(path, 'rb') as mask_img:
    analyze_images = service.analyze(collection_ids=[collection],
                                     features=[AnalyzeEnums.Features.OBJECTS.value],
                                     image_files=[FileWithMetadata(mask_img)]).get_result()

analyze_images

# 3. Visualise

obj = analyze_images['images'][0]['objects']['collections'][0]['objects'][0]['object']

from matplotlib import pyplot as plt

img = cv2.imread(path)

# %%

img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

# %%

font = cv2.FONT_HERSHEY_SIMPLEX

# %%

plt.imshow(img)

```

Figure 7: Connecting Watson Cloud Server to the code

```
num = ''
if obj == 'plastic':
    num = 'blue'
elif obj == 'metal':
    num = 'blue'
elif obj == 'glass':
    num = 'blue'
elif obj == 'cardboard':
    num = 'black'
elif obj == 'paper':
    num = 'black'
else:
    num = 'black'

return render_template("identify.html", object = obj, num = num)

if __name__ == "__main__":
    app.run(debug=True)
```

**Figure 8:** Decision mechanism

#### 6.1.1.2 HTML

This section outlines the code required using HTML coding to create the BeEco website.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>BeCo</title>

    <link rel="shortcut icon" type="image/jpg" href="static/images/Favicon.png"/>

    <!-- Google Fonts -->
    <link href="https://fonts.googleapis.com/css?family=Montserrat|Ubuntu" rel="stylesheet">

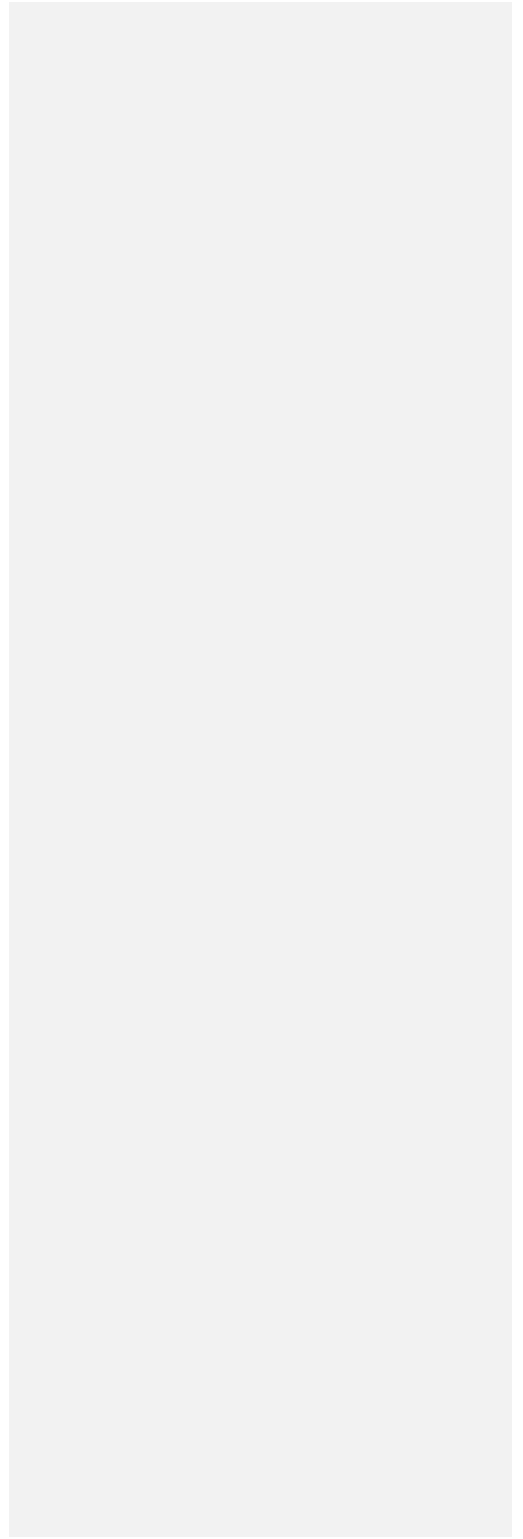
    <!-- CSS Stylesheets -->
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.8.0/css/bootstrap.min.css" integrity="sha384-S52xjxvr8" crossorigin="anonymous">
    <link rel="stylesheet" href="static/css/styles.css">

    <!-- Javascript -->

    <!-- Font Awesome -->
    <script defer src="https://use.fontawesome.com/releases/v5.8.7/js/all.js"></script>

    <!-- Bootstrap Scripts -->
    <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DK1IvYIK3iE92iG8K" crossorigin="anonymous"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js" integrity="sha384-ApBahg801K6MLYg3Op2h" crossorigin="anonymous"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.8.0/js/bootstrap.min.js" integrity="sha384-Pw6" crossorigin="anonymous"></script>
  </head>

  <body>
    <section class="colored-section" id="title">
      <div class="container-fluid">
        <!-- Nav Bar -->
        <nav class="navbar navbar-expand-lg navbar-dark">
          <a class="navbar-brand" href="">BeCo</a>
        </nav>
      </div>
    </section>
  </body>
</html>
```



```
<!-- Nav Bar -->
<nav class="navbar navbar-expand-lg navbar-dark">
  <a class="navbar-brand" href="#">BeCo/ </a>
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarTogglerDemo02">
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse" id="navbarTogglerDemo02">
    <ul class="navbar-nav ml-auto">
      <li class="nav-item">
        <a class="nav-link" href="#>Calendar/ </a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#>First Scan/ </a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#>Footer/ </a>
      </li>
    </ul>
  </div>
</nav>

<!-- Title -->
<div class="row">
  <div class="col-lg-6">
    <h3 class="h3-heading">I am a man, green, recycling machine/ </h3>
    <button type="button" class="btn btn-dark btn-lg download-button"><i class="fab fa-apple"></i> Download/ </button>
    <button type="button" class="btn btn-outline-light btn-lg download-button"><i class="fab fa-google-play"></i> Download/ </button>
  </div>
  <div class="col-lg-6">
    
</div>

<!-- Features -->
<section class="white-section" id="features">
  <div class="container-fluid">
    <div class="row">
      <div class="feature-box col-lg-4">
        <i class="icon fas fa-check-circle fa-4x"></i>
        <h3 class="feature-title">Easy to use.</h3>
        <p>>>So easy to use, very user friendly/</p>
      </div>
      <div class="feature-box col-lg-4">
        <i class="icon fas fa-wallet fa-4x"></i>
        <h3 class="feature-title">Versatility/</h3>
        <p>>>Works with eight different material types/</p>
      </div>
      <div class="feature-box col-lg-4">
        <i class="icon fas fa-wallet fa-4x"></i>
        <h3 class="feature-title">Guaranteed to work/</h3>
        <p>>>Discover materials with high efficiency/</p>
      </div>
    </div>
  </div>
</section>

<!-- Testimonials -->
<section class="colored-section" id="ps">
  <div id="testimonial-carousel" class="carousel slide" data-ride="false">
    <div class="carousel-inner">
      <div class="carousel-item active container-fluid">
```

```

<!-- Testimonials -->
<section class="colored-section" id="pc">
  <div id="testimonial-carousel" class="carousel slide" data-ride="false">
    <div class="carousel-inner">
      <div class="carousel-item active container-fluid">
        <h2 class="testimonial-text">A Points A </h2>
        </div>
        <div class="testimonial-text">
          <h2 class="testimonial-text">Calendar </h2>
          <em>Set your recycle dates</em>
        </div>
      </div>
      <div class="carousel-control-prev" href="#testimonial-carousel" role="button" data-slide="prev">
        <span class="carousel-control-prev-icon"></span>
      </div>
      <div class="carousel-control-next" href="#testimonial-carousel" role="button" data-slide="next">
        <span class="carousel-control-next-icon"></span>
      </div>
    </div>
  </section>

<!-- Scan -->
<section class="white-section" id="first">
  <a class="anchor" href="#"></a>
  <h2 class="section-heading">Scan your waste </h2>
</section>

<!-- Call to Action -->
<section class="colored-section" id="pac">
  <div class="container-fluid">
    <h2 class="big-heading">Recycling made better</h2>
    <p class="about-page-text">As a project to save recyclable items and improve recycling</p>
    <button class="download-button btn btn-lg btn-dark" type="button">Download</button>
    <button class="download-button btn btn-lg btn-light" type="button">Download</button>
  </div>
</section>

<!-- Footer -->
<footer class="white-section" id="footer">
  <div class="container-fluid">
    <div class="social-icon">
      <a href="#"></a>
      <a href="#"></a>
      <a href="#"></a>
      <a href="#"></a>
    </div>
    <p>© Copyright 2018 {year} Bafco</p>
  </div>
</footer>
</body>
</html>

```

Figure 9: Main page (HTML)

```
<!DOCTYPE html>
<html>
<link rel="stylesheet" href="static/css/styles.css">
<body style="background-color: #000000">

<a class="anchor" href="/scan" class="section-heading-2">Scan your waste</a>

<div class="video">
  <div id="container">
    <video autoplay="true" id="videoElement"></video>
    <script>
      var video = document.querySelector("#videoElement");

      if (navigator.mediaDevices.getUserMedia({ video: true })
        .then(function (stream) {
          video.srcObject = stream;
        })
        .catch(function (error) {
          console.log("Something went wrong!");
        })
      );
    </script>
  </div>
</div>

</div>
<div>
  <a class="anchor" href="/identify" class="pin"> See the Material </a>
</div>
</div>
```

Figure 10: Scan Page(HTML)

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Recycling Date</title>
  <link rel="stylesheet" href="/static/css/CalendarPicker.style.css">
</head>
<style>
  body {
    display: grid;
    justify-items: center;
    align-content: center;
    min-height: 180vh;
    padding: 0;
    margin: 0;
    font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen',
      'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',
      sans-serif;
  }

  #download-options {
    margin-bottom: 2rem;
  }

  #download-options h3:last-of-type {
    margin-top: 2rem;
  }

  #download-options code {
    font-size: 1.2rem;
  }

  #myCalendarWrapper {
    width: 48em;
    border: 5px solid #12064e;
    padding: 2em 4em;
    position: relative;
    font-size: 1rem;
    margin: 2rem auto 5rem;
  }

  code {
    font-size: 0.7rem;
    background-color: rgb(249, 249, 249);
    padding: 8px 8px;
    margin: 1rem 8px 1rem;
    border-radius: 5px;
    box-shadow: 2px 2px 0.2rem rgba(0, 0, 0, 0.2);
  }

  code h3 {
    padding: 0;
    margin: 0;
    font-size: 1.5rem;
    margin-bottom: 1rem;
    font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen',
      'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',
      sans-serif;
  }

  code pre {
    font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New', monospace;
  }

  .pink {
    color: rgb(233, 72, 99);
  }

  #example {
    margin: 0 auto;
    text-align: center;
    font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen',
      'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',
      sans-serif;
  }

  #example h3:not(:first-of-type) {
    margin-top: 3rem;
  }

  #example p {
    font-size: 0.9rem;
  }
</style>
```

```
#download-options h2:last-of-type {
  margin-top: 2rem;
}

#download-options code {
  font-size: 1.2rem;
}
</style>
</head>
<body>

<h3>Pick Your Recycling Date</h3>

<div id="showcase-wrapper">
  <div id="myCalendarWrapper"></div>
  <div id="example">
    <h3 class="rec">You will recycle on:</h3>
    <div class="rec" id="current-datestring"></div>
  </div>
  <div>
    <div style="color: #999999" id="current-date"></div>
    <div style="color: #999999" id="current-day"></div>
  </div>
</body>

<script src="/static/css/calendarPicker.js"></script>

<script>
  const nextYear = new Date().getFullYear() + 1;
  const myCalendar = new CalendarPicker({myCalendarWrapper}, {
    // If max < min or min > max then the only available day will be today.
    min: new Date(),
    max: new Date(nextYear, 10) // NOTE: new Date(nextYear, 10) is "Sun Nov 01 <nextYear>"
  });

  const currentDateElement = document.getElementById('current-date');
  currentDateElement.textContent = myCalendar.value;

  const currentDayElement = document.getElementById('current-day');
  currentDayElement.textContent = myCalendar.value.getDay();

  const currentDateString = document.getElementById('current-datestring');
  currentDateString.textContent = myCalendar.value.toString();

  myCalendar.onValueChange((currentValue) => {
    currentDateElement.textContent = currentValue;
    currentDayElement.textContent = currentValue.getDay();
    currentDateString.textContent = currentValue.toString();

    console.log('The current value of the calendar is: ', {currentValue});
  });
</script>
</html>
```

Figure 11: Calendar Page(HTML)



```
<!DOCTYPE html>
<html>
<link rel="stylesheet" href="static/css/styles.css">
<body style="background-color: #d8ebe4">

  <div>
    <p class = "par">Your Materials is: {{object}} </p>
    <p class = "par">Use {{num}} bin</p>
  </div>

</body>
</html>
```

**Figure 12:** Result Page(HTML)

#### 6.1.1.3 CSS Code

The CSS Code is outlined below, it is used to change the look of the material provided by the HTML code.

```
body {
  font-family: "Montserrat";
  text-align: center;
}

h1, h2, h3, h4, h5, h6 {
  font-family: "Montserrat-Bold";
}

p {
  color: #8f8f8f;
}

/* Headings */

.big-heading {
  font-family: "Montserrat-Black";
  font-size: 3.5rem;
  line-height: 1.5;
}

.section-heading {
  font-size: 3rem;
  line-height: 1.5;
}

/* Containers */

.container-fluid {
  padding: 7% 15%;
}

/* Sections */

.colored-section {
  background-color: #fed049;
  color: #fff;
}
```

```
.white-section {
  background-color: #fff;
}

/* Navigation Bar */

.navbar {
  padding: 0 0 4.5rem;
}

.navbar-brand {
  font-family: "Ubuntu";
  font-size: 2.5rem;
  font-weight: bold;
}

.nav-item {
  padding: 0 18px;
}

.nav-link {
  font-size: 1.65rem;
  font-family: "Montserrat-Light";
}

/* Buttons */

.download-button {
  margin: 5% 3% 5% 0;
}

.section-heading-2 {
  font-size: 3rem;
  line-height: 1.5;
  color: #007580;
}

.par {
  font-size: 2.5rem;
  line-height: 1.5;
  color: #007580;
}
```

```
/* Title Section */

#title {
  background-color: #007580;
  color: #fff;
  text-align: left;
}

#title .container-fluid {
  padding: 3% 15% 7%;
}

/* Title Image */

.title-image {
  width: 78%;
  transform: rotate(25deg);
  position: absolute;
  right: 10%;
  animation: rotatelogo infinite 20s linear;
}

.about-page{
  color: #282846;
  font-size: 125%;
}

/* Features Section */

#features {
  position: relative;
}

.feature-title {
  font-size: 1.5rem;
}

.feature-box {
  padding: 4.5%;
}
```

```
.feature-box {
  padding: 4.5%;
}

.icon {
  color: #fed049;
  margin-bottom: 1rem;
}

.icon:hover {
  color: #ac0d0d;
}

/* Testimonial Section */

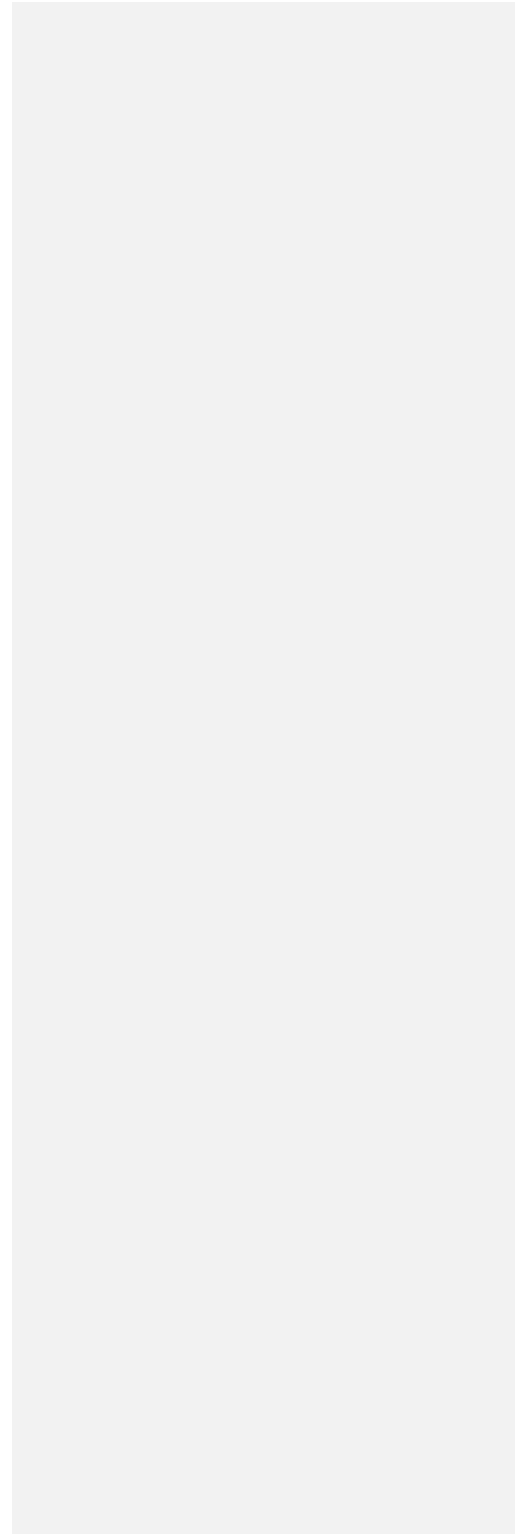
#testimonials {
  background-color: #007580;
}

.testimonial-text {
  font-size: 3rem;
  line-height: 1.5;
}

.testimonial-image {
  width: 10%;
  border-radius: 100%;
  margin: 20px;
}

#press {
  background-color: #fed049;
  padding-bottom: 3%;
}

.press-logo {
  width: 15%;
  margin: 20px 20px 50px;
}
```



```
#first {  
  padding: 100px;  
}  
  
.price-text {  
  font-size: 3rem;  
  line-height: 1.5;  
}  
  
.pricing-column {  
  padding: 3% 2%;  
}  
  
/* CTA Section */  
  
/* Footer Section */  
  
.social-icon {  
  margin: 20px 10px;  
}  
  
.Webcam{  
  margin-top: 50px;  
  align-content: center;  
}  
  
.anchor{  
  color: white;  
}  
  
.anchor2{  
  color: #007580;  
}  
  
@media (max-width: 1028px) {  
  
  #title {  
    text-align: center;  
  }  
}
```

```
    text-align: center;
  }
}

.title-image {
  position: static;
  transform: rotate(0);
}
}
}
@keyframes rotatelogo{
  from{
    transform: rotate(0deg);
  }
  to{
    transform: rotate(360deg);
  }
}
}

.loader {
  margin-left: 750px;
  border: 16px solid #f3f3f3; /* Light grey */
  border-top: 16px solid blue;
  border-radius: 50%;
  width: 120px;
  height: 120px;
  animation: spin 2s linear infinite;
}

@keyframes spin {
  0% { transform: rotate(0deg); }
  100% { transform: rotate(360deg); }
}
```

```

* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: "Quicksand", sans-serif;
}

html {
  font-size: 62.5%;
}

.container {
  width: 100%;
  height: 100vh;
  background-color: #fed049;
  color: #d8ebe4;
  display: flex;
  justify-content: center;
  align-items: center;
}

.calendar {
  width: 45rem;
  height: 52rem;
  background-color: #282846;
  box-shadow: 0 0.5rem 3rem rgba(0, 0, 0, 0.4);
}

.month {
  width: 100%;
  height: 12rem;
  background-color: #007580;
  display: flex;
  justify-content: space-between;
  align-items: center;
  padding: 0 2rem;
  text-align: center;
  text-shadow: 0 0.3rem 0.5rem rgba(0, 0, 0, 0.5);
}

.month i {
  font-size: 0.5em;
}

```



```
.month i {
  font-size: 2.5rem;
  cursor: pointer;
}

.month h1 {
  font-size: 3rem;
  font-weight: 400;
  text-transform: uppercase;
  letter-spacing: 0.2rem;
  margin-bottom: 1rem;
}

.month p {
  font-size: 1.6rem;
}

.weekdays {
  width: 100%;
  height: 5rem;
  padding: 0 0.4rem;
  display: flex;
  align-items: center;
}

.weekdays div {
  font-size: 1.5rem;
  font-weight: 400;
  letter-spacing: 0.1rem;
  width: calc(44.2rem / 7);
  display: flex;
  justify-content: center;
  align-items: center;
  text-shadow: 0 0.3rem 0.5rem rgba(0, 0, 0, 0.5);
}

.days {
  width: 100%;
  display: flex;
  flex-wrap: wrap;
  padding: 0 2rem;
```

```
.days {
  width: 100%;
  display: flex;
  flex-wrap: wrap;
  padding: 0.2rem;
}

.days div {
  font-size: 1.4rem;
  margin: 0.3rem;
  width: calc(40.2rem / 7);
  height: 5rem;
  display: flex;
  justify-content: center;
  align-items: center;
  text-shadow: 0 0.3rem 0.5rem rgba(0, 0, 0, 0.5);
  transition: background-color 0.2s;
}

.days div:hover:not(.today) {
  background-color: #262626;
  border: 0.2rem solid #777;
  cursor: pointer;
}

.prev-date,
.next-date {
  opacity: 0.5;
}

.today {
  background-color: #007580;
}

.calendar-text{
  position: absolute;
  margin-right: 150px;
  margin-top: 600px;
  text-align: center;
}
```

```

body{
  background-color: #f9f7f7;
}

h1{
  color: #3f72af
}

.rec{
  font-size: 1.5rem;
  color: #3f72af;
}

.recp{
  font-size: 1.5rem;
  color: #112d4e;
}

#calendar-wrapper {
  width: 100%;
  display: grid;
  grid-template-rows: 2.5em 1fr;
  font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue', sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  box-sizing: border-box;
  font-size: inherit;
  color: #3f72af
}

#calendar-wrapper header {
  grid-row: 1/2;
  font-size: 2em;
  box-sizing: inherit;
}

#calendar-wrapper header h1 {

```

```
#calendar-wrapper header h4 {
  padding: 0;
  margin: 0;
}

#calendar {
  display: grid;
  grid-template-columns: 1fr 4em;
  grid-template-rows: 3em 1fr;
  box-sizing: inherit;
  padding: 0;
  margin: 0;
}

#calendar #calendar-days,
#calendar #calendar-grid {
  display: grid;
  grid-template-columns: repeat(7, 1fr);
  align-items: center;
  padding: 0 1em;
  box-sizing: inherit;
  margin: 0;
}

#calendar #calendar-days {
  justify-items: center;
  grid-row: 1/2;
  grid-column: 1/2;
}

#calendar #calendar-grid {
  grid-template-rows: repeat(6, 1fr);
  justify-items: flex-start;
  grid-row: 2/-1;
  grid-column: 1/2;
  width: 100%;
}

#calendar #calendar-grid > time {
  display: flex;
  align-items: center;
```

```

#calendar #calendar-grid {
  grid-template-rows: repeat(6, 1fr);
  justify-items: flex-start;
  grid-row: 2/-1;
  grid-column: 1/2;
  width: 100%;
}

#calendar #calendar-grid > time {
  display: flex;
  align-items: center;
  justify-content: center;
  cursor: pointer;
  margin: 0.4em;
  padding: 1em 0.2em;
  width: 90%;
  background-color: rgba(108, 108, 108, 0.02);
  border-radius: 3px;
  border: 2px solid transparent;
  transition: box-shadow 200ms, border 100ms;
  box-shadow: 1px 1px rgba(0, 0, 0, 0);
  box-sizing: inherit;
}

#calendar #calendar-grid time.selected,
#calendar #calendar-grid time.selected:focus {
  outline: none;
  border: 2px solid #c64949;
  box-sizing: inherit;
}

#calendar #calendar-grid time.disabled,
#calendar #calendar-grid time.disabled:focus {
  outline: none;
  border: 2px solid transparent;
  box-sizing: inherit;
  cursor: not-allowed;
  background-color: rgba(30, 39, 51, 0.2);
  box-shadow: 1px 1px rgba(0, 0, 0, 0);
}

```

```

#calendar #calendar-grid time.disabled,
#calendar #calendar-grid time.disabled:focus {
  outline: none;
  border: 2px solid transparent;
  box-sizing: inherit;
  cursor: not-allowed;
  background-color: rgba(30, 39, 51, 0.2);
  box-shadow: 1px 1px rgba(0, 0, 0, 0);
}

#calendar #calendar-grid > *:hover {
  border: 2px solid rgba(72, 72, 72, 0.5);
  box-shadow: 1px 1px rgba(0, 0, 0, 0.2);
  box-sizing: inherit;
}

#calendar #navigation-wrapper {
  display: flex;
  flex-direction: column;
  justify-content: space-between;
  align-items: center;
  grid-row: 2/-1;
  grid-column: 2/-1;
  box-sizing: inherit;
  padding: 1em 0 3.5em;
  margin: 0;
}

#calendar #navigation-wrapper button {
  cursor: pointer;
  background: transparent;
  border: 0;
  font-size: 3em;
  width: 1.6em;
  box-sizing: inherit;
  padding: 0.1em;
  margin: 0;
}

#calendar #navigation-wrapper button:hover {
  opacity: 0.75;
}

```

```
    grid-row: 2/-1;
    grid-column: 2/-1;
    box-sizing: inherit;
    padding: 1em 0 3.5em;
    margin: 0;
}

#calendar #navigation-wrapper button {
    cursor: pointer;
    background: transparent;
    border: 0;
    font-size: 3em;
    width: 1.6em;
    box-sizing: inherit;
    padding: 0.1em;
    margin: 0;
}

#calendar #navigation-wrapper button:hover {
    opacity: 0.75;
}

#calendar #navigation-wrapper button svg {
    margin: 0;
    padding: 0;
    width: 100%;
    height: 1.6em;
    display: block;
    box-sizing: inherit;
}

#calendar #navigation-wrapper button:first-of-type {
    transform: rotate(180deg);
}
```

Figure 13: CSS Code to create BeEco

#### 6.1.1.4 Java Script

This section outlines the use of Java Script to create the BeEco website. JavaScript was used for the functionality of the Calendar. For example, changing color and date when calendar is clicked.

```
// Element to insert calendar in.
this.userElement = document.querySelector(element);

// Setting current date as readable text.
this._setDateText();

// The elements used to build the calendar.
this.calendarWrapper = document.createElement('div');
this.calendarElement = document.createElement('div');
this.calendarHeader = document.createElement('header');
this.calendarHeaderTitle = document.createElement('h4');
this.navigationWrapper = document.createElement('section');
this.previousMonthArrow = document.createElement('button');
this.nextMonthArrow = document.createElement('button');
this.calendarGridDays = document.createElement('section');
this.calendarGrid = document.createElement('section');
this.calendarDayElementType = 'time';

// Hard-coded list of all days.
this.listOfAllDaysAsText = [
  'Monday',
  'Tuesday',
  'Wednesday',
  'Thursday',
  'Friday',
  'Saturday',
  'Sunday'
];

// Hard-coded list of all months.
this.listOfAllMonthsAsText = [
  'January',
  'February',
  'March',
  'April',
  'May',
  'June',
  'July',
  'August',
  'September',
  'October',
  'November',
  'December'
];
```



```

// Polyfill for Element.prototype.closest (for IE 9)
if (!Element.prototype.matches) { Element.prototype.matches = Element.prototype.msMatchesSelector || Element.prototype.webkitMatchesSelector; }
if (!Element.prototype.closest) {
  Element.prototype.closest = function(s) {
    var el = this;
    do {
      if (Element.prototype.matches.call(el, s)) return el;
      el = el.parentElement || el.parentNode;
    } while (el != null && el.nodeType !== 1);
    return null;
  };
}

/**
 * @class CalendarPicker.
 * @description Provides a simple way to get a minimalistic calendar in your DOM.
 * @author Mathias Picker - 29 July 2020.
 */
function CalendarPicker(element, options) {
  // Core variables.
  this.date = new Date();
  this._formatDateToInit(this.date);

  this.day = this.date.getDay();
  this.month = this.date.getMonth();
  this.year = this.date.getFullYear();

  // Storing the todays date for practical reasons.
  this.today = this.date;

  // The calendars value should always be the current date.
  this.value = this.date;

  // Ranges for the calendar (optional).
  this.min = options.min;
  this.max = options.max;
  this._formatDateToInit(this.min);
  this._formatDateToInit(this.max);

  // Element to insert calendar in.
  this.userElement = document.querySelector(element);
}

```

```
this.listOfAllMonthsAsText = [
    'January',
    'February',
    'March',
    'April',
    'May',
    'June',
    'July',
    'August',
    'September',
    'October',
    'November',
    'December'
];

// Creating the calendar
this.calendarWrapper.id = 'calendar-wrapper';
this.calendarElement.id = 'calendar';
this.calendarGridDays.id = 'calendar-days';
this.calendarGrid.id = 'calendar-grid';
this.navigationWrapper.id = 'navigation-wrapper';
this.previousMonthArrow.id = 'previous-month';
this.nextMonthArrow.id = 'next-month';

this._insertHeaderIntoCalendarWrapper();
this._insertCalendarGridDaysHeader();
this._insertDaysIntoGrid();
this._insertNavigationButtons();
this._insertCalendarIntoWrapper();

this.userElement.appendChild(this.calendarWrapper);
}

/**
 * @param {Number} The month number, 0 based.
 * @param {Number} The year, not zero based, required to account for leap years.
 * @return {Array<Date>} List with date objects for each day of the month.
 * @author Juan Mendes - 30th October 2012.
 */
CalendarPicker.prototype._getDaysInMonth = function (month, year) {
```

```

CalendarPicker.prototype._getDaysInMonth = function (month, year) {
  if ((!(month && month !== 0) || !(year && year !== 0)) return;

  var date = new Date(year, month, date, 1);
  var days = [];

  while (date.getMonth() === month) {
    days.push(new Date(date));
    date.setDate(date.getDate() + 1);
  }

  return days;
}

/**
 * @param {Date|Year} date.
 * @description Sets the clock of a date to 00:00:00 to be consistent.
 */
CalendarPicker.prototype._formatDateToInit = function (date) {
  if (!date) return;
  date.setHours(0, 0, 0);
}

/**
 * @description Sets the current date as readable text in their own variables
 */
CalendarPicker.prototype._setDataText = function () {
  // Setting current date as readable text.
  var dateData = this.date.toString().split(separator);
  this.dayAsText = dateData[0];
  this.monthAsText = dateData[1];
  this.dateAsText = dateData[2];
  this.yearAsText = dateData[3];
}

/**
 * @description Inserts the calendar into the wrapper and adds eventListeners for the calendar-grid.
 */
CalendarPicker.prototype._insertCalendarIntoWrapper = function () {
  this.calendarWrapper.appendChild(this.calendarElement);
}

```

```

/**
CalendarPicker.prototype._insertCalendarIntoWrapper = function () {
  this.calendarWrapper.appendChild(this.calendarElement);

  /**
   * @param {Event} event An event from an eventListener.
   */
  var handleSelectedElement = (event) => {
    if (event.target.nodeName.toLowerCase() === this.calendarDayElementType && !event.target.classList.contains('disabled')) {
      // Removes the 'selected' class from all elements that have it.
      Array.from(document.querySelectorAll(selectors.selected)).forEach(element => element.classList.remove('selected'));

      // Adds the 'selected'-class to the selected date.
      event.target.classList.add('selected');

      this.value = event.target.value;

      // Fires the onChange function with the provided callback.
      this.onChange(this.callback);
    }
  };

  this.calendarGrid.addEventListener('click', handleSelectedElement, {options: false});

  this.calendarGrid.addEventListener('keydown', {listener: (keyEvent: KeyboardEvent) => {
    if (keyEvent.key !== 'Enter') return;
    handleSelectedElement(keyEvent);
  }, options: false});
}

/**
 * @description Adds the "main" calendar-header.
 */
CalendarPicker.prototype._insertHeaderIntoCalendarWrapper = function () {
  this.calendarHeaderTitle.textContent = this.listOfAllMonthsAsText[this.month] + ' - ' + this.year;
  this.calendarHeader.appendChild(this.calendarHeaderTitle);
  this.calendarWrapper.appendChild(this.calendarHeader);
}

```

```

/**
 * @description Inserts the calendar-grid header with all the weekdays.
 */
CalendarPicker.prototype._insertCalendarGridDaysHeader = function () {
  this.listOfAllDaysAsText.forEach(day => {
    var dayElement = document.createElement( tagName: 'span' );
    dayElement.textContent = day;
    this.calendarGridDays.appendChild( dayElement );
  });

  this.calendarElement.appendChild( this.calendarGridDays );
}

/**
 * @description Adds the "Previous" and "Next" arrows on the side-navigation.
 * Also inits the click-events used to navigating.
 */
CalendarPicker.prototype._insertNavigationButtons = function () {
  // Ugly long string, but at least the svg is pretty.
  var arrowSvg = '<svg enable-background="new 0 0 386.257 386.257" viewBox="0 0 386.257 386.257" xmlns="http://www.w3.org/2008/svg"><path d="m0 96

  this.previousMonthArrow.innerHTML = arrowSvg;
  this.nextMonthArrow.innerHTML = arrowSvg;

  this.previousMonthArrow.setAttribute( qualifiedName: 'aria-label', value: 'Go to previous month' );
  this.nextMonthArrow.setAttribute( qualifiedName: 'aria-label', value: 'Go to next month' );

  this.navigationWrapper.appendChild( this.previousMonthArrow );
  this.navigationWrapper.appendChild( this.nextMonthArrow );

  // Cannot use arrow-functions for IE support :(
  var that = this;
  this.navigationWrapper.addEventListener( type: 'click', listener: function (clickEvent :MouseEvent) {
    if (clickEvent.target.closest( '#' + that.previousMonthArrow.id ) ) {
      if (that.month == 0) {
        that.month = 11;
        that.year -- 1;
      } else {
        that.month -- 1;
      }
    }
  });
}

```

```

this.navigationWrapper.addEventListener({ type: 'click', listener: function (clickEvent :MouseEvent) {
  if (clickEvent.target.closest("#" + that.previousMonthArrow.id) ) {
    if (that.month == 0) {
      that.month = 11;
      that.year -- 1;
    } else {
      that.month -- 1;
    }
    that._updateCalendar();
  }

  if (clickEvent.target.closest("#" + that.nextMonthArrow.id) ) {
    if (that.month == 11) {
      that.month = 0;
      that.year ++ 1;
    } else {
      that.month ++ 1;
    }
    that._updateCalendar();
  }
}, options: false)

that.calendarElement.appendChild(that.navigationWrapper);
}

/**
 * @description Adds all the days for current month into the calendar-grid.
 * Takes into account which day the month starts on, so that "empty/placeholder" days can be added
 * in case the month for example starts on a Thursday.
 * Also disables the days that are not within the provided.
 */
CalendarPicker.prototype._insertDaysIntoGrid = function () {
  this.calendarGrid.innerHTML = '';

  var arrayOfDay = this._getDaysInMonth(this.month, this.year);
  var firstDayOfMonth = arrayOfDay[0].getDay();

  // Converting Sunday (0 when using getDay()) to 7 to make it easier to work with.
  firstDayOfMonth = firstDayOfMonth == 0 ? 7 : firstDayOfMonth;

  if (1 < firstDayOfMonth) {

```

```

* Takes into account which day the month starts on, so that "empty/placeholder" days can be added
* in case the month for example starts on a Thursday.
* Also disables the days that are not within the provided.
*/
CalendarPicker.prototype._insertDaysIntoGrid = function () {
    this.calendarGrid.innerHTML = '';

    var arrayOfDay = this._getDaysInMonth(this.month, this.year);
    var firstDayOfMonth = arrayOfDay[0].getDay();

    // Converting Sunday (0 when using getDay()) to 7 to make it easier to work with.
    firstDayOfMonth = firstDayOfMonth === 0 ? 7 : firstDayOfMonth;

    if (1 < firstDayOfMonth) {
        arrayOfDay = Array(arrayLength: firstDayOfMonth - 1).fill({value: false, start: 0}).concat(arrayOfDay);
    }

    arrayOfDay.forEach(date => {
        var dateElement = document.createElement({tagName: date ? this.calendarDayElementType : 'span'});
        var Date = date.toString().split('separator: ')[2];

        var dateIsTheCurrentValue = this.value.toString() == date.toString();
        if (dateIsTheCurrentValue) this.activeDateElement = dateElement;

        var dateIsBetweenAllowedRange = (this.min || this.max) && (date.toString() != this.today.toString() && (date < this.min || date > this.max));
        if (dateIsBetweenAllowedRange) {
            dateElement.classList.add('disabled');
        } else {
            dateElement.tabIndex = 0;
            dateElement.value = date;
        }

        dateElement.textContent = date ? Date : '';
        this.calendarGrid.appendChild(dateElement);
    });

    this.calendarElement.appendChild(this.calendarGrid);
    this.activeDateElement.classList.add('selected');
}
/!

```

```
    this.calendarElement.appendChild(this.calendarGrid);
    this.activeDateElement.classList.add('selected');
}

/**
 * @description Updates the core-values for the calendar based on the new month and year
 * given by the navigation. Also updates the UI with the new values.
 */
CalendarPicker.prototype._updateCalendar = function () {
    this.date = new Date(this.year, this.month);

    this._setDateText();

    this.day = this.date.getDay();
    this.month = this.date.getMonth();
    this.year = this.date.getFullYear();

    // Cannot use arrow-functions for IE support :(
    var that = this;
    window.requestAnimationFrame( callback: function () {
        that.calendarHeaderTitle.textContent = that.listOfAllMonthsAsText[that.month] + ' - ' + that.year;
        that._insertDaysIntoGrid();
    })
}

/**
 * @param {Function} callback
 * @description A "listener" that lets the user do something everytime the value changes.
 */
CalendarPicker.prototype.onValueChange = function (callback) {
    if (this.callback) return this.callback(this.value);
    this.callback = callback;
}
```

Figure 14: How to use Java Script to use BeEco

## 6.2 Prototype 2

The main focus of prototype two was to get the material detection working. Using IBM Watson Visual Recognition software this was accomplished. This software allows for a database of images to be created and for them to be categorized into the material that they are made of. Each material has 250 images that had been uploaded into the database, trained into the system and sorted by the team into their corresponding material classes. To summarize, using the training feature integrated into IBM we can train the database in order to determine the material being scanned.

## 6.3 Prototype 3

The third prototype focused on getting the subsystem of the bin identification system up and properly running to fulfill the client's needs and wants. For this prototype there were also two sub focuses, finalizing the user interface by making it look more organized and user-friendly as well ensuring that the calendar function was functioning.



Additionally, some improvements were made to the image recognition database on IBM Watson. This prototype resulted in the calendar function being a manual input where the user inputs the day of their recycling pick up and the site will remember and store that information, as well as achieving accurate recycling bin sorting.

#### 6.4 BOM (Bill of Materials)

The BOM documents all software's that were used in the creation of this application (BeEco) specifically.

**Table #3: Bill of Materials (BOM)**

Material	Use	Cost
IBM Watson Visual Recognition	Used to train the website using image recognition and artificial intelligence, System of recognizing the type of material.	Free / 1000 tests per month

Though there were no costs associated with the project that was developed in the group IBM Watson only allowed for 1000 tests a month, this means that more would have to be purchased if this application were to be rolled out on a large scale, a domain would also have to be purchased to make it public as well as patents or trademarks for the BeEco brand.

##### 6.4.1 Equipment list

Though the group already had all equipment on hand it should also be considered that there are other materials that are necessary to obtain in order to operate, update and/or recreate the project.

**Table #4: Equipment List and Details on the Requirements or Use**

Equipment	Details
Computer	With decent processing speed, RAM, graphics and adequate storage.
Hightspeed Wi-Fi connection	Avoids failures or crashing when uploading on IBM Watson
General Knowledge of	HTML5, CSS, JS, Python and website design/creation
IBM Watson account	One for the group of individuals is adequate
Webcam (if no camera on computer)	To test the system and scan materials
Recyclable Materials	Used for scanning and testing stages if applicable

## 6.5 Testing & Validation

While iterating the prototypes to create the final product there were many stages of testing and different metrics that were tested at different stages. The process is outlined in this section for each of the prototypes the third being the final product.

### 6.5.1 Testing of Prototype 1

Once the first prototype was finished, we tested it and gathered feedback from peers on what could be improved. The prototype was tested to ensure its functionality, accuracy, and its user friendliness. The first test was to check and see if the program was able to open the camera, and it was. The second test was to rate the user friendliness on a scale of 1 to 10 with a passing value of 7 and an ideal value of 10. In order to check and see if the prototypes user interface was acceptable, we asked for feedback from peers and the team also gave it a score. The average score was 7.6 which is a passing score, however, it demonstrated that some adjustments should be made. Presented below in Table #5 are the compiled results from the tests.

Commented [GW5]: add figure numbers

**Table #5:** Beta Testing of the First Prototype, Testing the User-friendliness Metric

<b>Names</b>	<b>Rating out of 10</b>	<b>Feedback</b>	<b>Recommend to a friendly (yes or no)</b>
Joe	8	The software works fast, and it can open the camera. I really like the app.	yes
April	6	The camera is a bit slow, and the quality of the camera seems deteriorated.	no
Mark	9	Perfect app for the current stage in the app development I think this app	yes

		will be very successful.	
Linsay	7	The app looks very nice and its relatively quick in transitioning from screens.	yes
Emma	8	Overall, the app is very nice, and it successfully opens the camera, I had a good user experience.	yes

### 6.3.2 Testing of Prototype 2

The test we ran during the prototype development was a binary yes or no if the website was able to determine the material. The scanned items had an accuracy of around 80-90%.

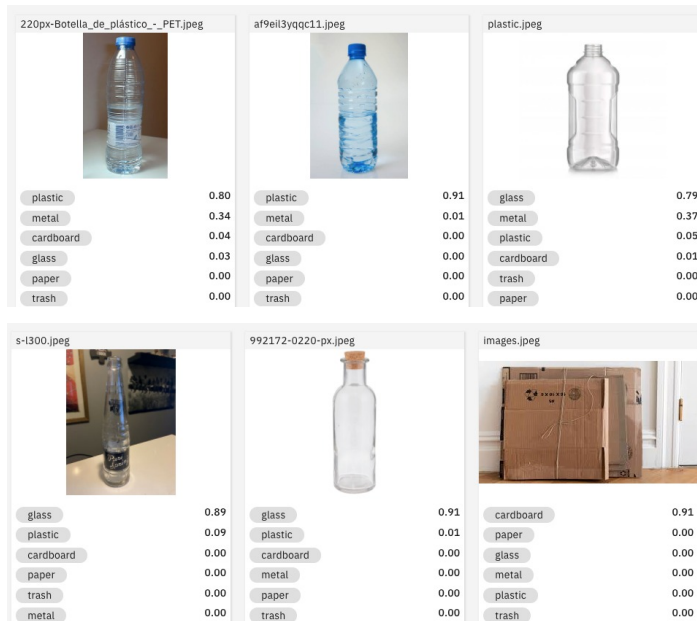
**Table #6: Beta Testing of the Second Prototype, Testing the Identification of Recyclable Material (Client Need)**

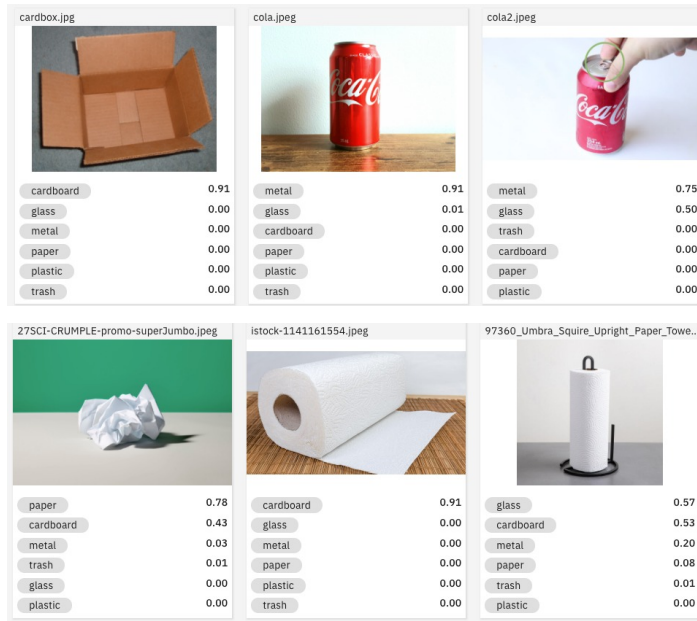
Name	Feedback	Rating (out of 10)	Identifies object and Recyclable material (Y/N)
Samantha	The app was working very well I liked the bright colors and all the available features. It works fairly accurately	8	Y
Greg	The app is really user friendly and correctly identifies the object that I am scanning	9	Y

<b>Lux</b>	I love the app and I think that it's very straight forward and easy to use to scan products to recycle	<b>9</b>	<b>Y</b>
------------	--	----------	----------

Alpha testing results were discussed as a group and documented in Project Deliverables; beta testing was also performed to gain feedback without bias of the group. For the second prototype the feedback was overall good, given that a sticking point was the mild inaccuracies this was a goal for future development and prototyping.

Below are figures representing the accuracy of each specific material and the compiled feedback on the user interface.





**Figure #15:** Results and Accuracies of Testing Recyclable Materials Using the IBM Watson Visual Recognition.

Here it shows IBM Watson as it matches the material it was given (shown in the images) with the images in the database. It is important to note that though there were some inaccuracies and that it is not 100% certain the materials such as glass and metal that it may get confused are placed in the same recycling bin, thus it would still accurately sort the recycling

### 6.3.3 Testing of Prototype 3

The testing we did on the third prototype was a binary yes or no if the scanner accurately depicts the bin to place the material in. This testing was performed by showing the BeEco website the material just as a user would and then clicking "See Material". The accuracy was recorded as multiple materials were tested. The accuracies of our scanned materials are tabulated below.

**Table #7: Results of Alpha Testing Prototype Three, Using IBM Watson and BeEco**

Material Scanned	Average System Accuracy
Glass	90 %
Cardboard	90%
Metal	85%
Plastic	83%
Paper	7%

As seen above the main area to improve upon for the BeEco System was the paper section. This being said, its main issue was confusing paper and cardboard which in Ottawa both end up in the Black bin. This once again means that the system would still identify the correct bin in which to throw away your material, however, additional images of paper should be added to IBM Watson in order to grow its paper database and increase its efficiency.

## 7. Conclusions and Recommendations for Future Work

Throughout the course of this project, our team was introduced to some valuable new concepts and experiences that taught us an important lesson in design related work for engineering courses.

The team learned how to cooperate and function as a team to efficiently accomplish tasks that were assigned to our group. We were introduced to new software's and applications such as Unity, TensorFlow, and IBM software, of which we ended up using IBM software in our final product. We also learned how to analyze and benchmark products readily available on the market in order to determine the proper characteristics of our final product.

For the future of the product, BeEco, the team planned to have a feedback loop implemented into the system. This feedback loop would ask the user if it had correctly identified the material that it scanned, if BeEco did not properly identify the material it would provide the user with the ability to tell the system what it is. This would ideally allow for the system to become more accurate and for the database to be increased as it learns more with more scanning inputs. In this feedback loop the team ideally purposes that the system should as the user if the material has been cleaned before Recycling it to promote proper recycling. There were also hopes to implement a user account system into the app with an option to enable cookies to provide users with the best personalized experience within the app. Should the product be successful in capturing the interests of our guests during design day, then ideally sponsors or prizes for our product would be found to provide users with prizes to put their points towards.

If the team had more time to developpe this project, then a fully-fledged application that can run on people's phones (not just a bookmarked website) would have been developed. This application would have included the feedback loop as well as have accounts or cookies to store peoples recycling pickup and points information. Finally, this product would also include having

prizes and coupons ready to be won using the point system. However, due to the time constraint of this semester, our product was instead developed as a website with a focus on the scanning function and sorting system.

Overall, the group was able to develop BeEco to solve the client's problem and meet his needs and wants of having an application that could identify and sort recycling in a cost effective, user-friendly, fun and versatile way.

## APPENDICES

### APPENDIX I: Design Files

This document acts as a summary of the documents outlined in the table below. The user manual largely focuses on how to create the final product with a lesser focus on the results and process of developing the lower feasibility or less functioning prototypes iterated in order to achieve the final product.

**Table #8: Referenced Documents**

Document Name	Document Location and/or URL	Issuance Date
PD A	Maker Repo	24/01/21
PD B	Maker Repo	31/01/21
PD C	Maker Repo	07/02/21
PD D	Maker Repo	21/02/21
PD E	Maker Repo	28/02/21
PD F	Maker Repo	07/02/21
PD G	Maker Repo	21/02/21
PD H	Maker Repo	28/02/21
Gantt Chart	<a href="#">Screenshot link</a>	NA

PD A consisted of a team contract to ensure that the team met deadlines, delivered high quality deliverables and prototypes and that team members were to be focused and held accountable for the lack of efforts or participation.

PD B followed the first client meeting and summarized the client's needs and wants, here the problem was also identified, and the problem statement was determined. The needs and wants were used to identify a set of metrics and benchmarking was performed to determine the competition or other products on the market.

PD C developed a set of design criteria based on the metrics and needs and wants from PD B. After the design criteria was further developed more research and benchmarking were done to determine key features to include in the product to be developed.

PD D summarizes the brainstorming process as well as ideas explored for the various subsystems of the product. Here a feasibility study was performed as well as a selection matrix was created in order to determine the best concept to move forward with.

PD E contained the BOM for the chosen concept. It also included sketches and a detailed design and outline of what the product would look like and what functions or subsystems it would contain. In this document, the testing and prototyping plans and schedules were also developed. This was to ensure that the team could meet the client's needs and wants, solve the problem and turn in all required work on time.

PD F outlined the first prototype, giving information on what was being tested and prototyped, the stopping criteria as well as the results for the given testing.

PD G outlined the second prototype, giving information on what was being tested and prototyped, the stopping criteria as well as the results for the given testing. This prototype also had a focus on explaining the team's logo and name for the product as well as some potential for marketing.



PD H outlined the third and final prototype, giving information on what was being tested and prototyped, the stopping criteria as well as the results for the given testing.

The Gantt chart was a project management planning and development tool developed using Wrike. This allowed the team to delegate and manage tasks efficiently as well as to ensure that deadlines were met.

