

6.2 Application Performance

Real-Time Interaction: The responsiveness to user input is critical for an engaging experience. Minimize the latency between user actions (such as selecting customization options) and visual feedback. Use asynchronous loading techniques to ensure smooth interaction even when loading complex models or textures.

Rendering Quality and Speed: High-quality rendering of 3D models is essential for visual fidelity. However, it should not compromise performance. Optimize rendering techniques such as level-of-detail (LOD) rendering and occlusion culling to improve frame rates. Implement efficient shaders and textures to reduce GPU load without sacrificing visual quality.

Network Performance: If the application requires online content or updates, optimize network communication to minimize latency and bandwidth usage. Implement efficient data synchronization strategies to ensure smooth multi-player experiences if applicable.

UI/UX Optimization: Design a streamlined user interface that minimizes clutter and simplifies navigation. Optimize UI rendering to minimize overhead and ensure smooth transitions between screens and menus. Provide clear visual feedback to indicate loading or processing tasks to prevent user frustration.

Reliability and Stability: Ensure the application is robust and resilient to prevent crashes or downtime during user interactions. Implement error handling and recovery mechanisms to gracefully handle exceptions and failures. Conduct thorough testing, including stress testing and load testing, to identify and address potential failure points. Monitor application performance and health in real-time, with automated alerts for anomalies or issues that require attention.

6.3 Code

6.3.1 app.py

```

1 import os
2 from flask import Flask, request, jsonify
3 from flask_cors import CORS # Import the CORS module
4 import asyncio
5 from gradio_client import Client
6 import gradio_client
7 import shutil
8
9 client = Client("stabilityai/TripoSR")
10 # Create a Flask app instance
11 app = Flask(__name__)
12 CORS(app) # Enable CORS for all routes
13
14 # Define the directory to store uploaded images
15 UPLOAD_FOLDER = 'images'
16 if not os.path.exists(UPLOAD_FOLDER):
17     os.makedirs(UPLOAD_FOLDER)
18
19 # Set the upload folder for Flask app
20 app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
21
22 # Define a route for the root URL

```

```

23 @app.route('/')
24 def hello():
25     return 'Server is running...'
26
27 # Define an endpoint to upload images
28 @app.route('/upload', methods=['POST'])
29 def upload_image():
30     # Check if the POST request contains a file
31     if 'file' not in request.files:
32         return jsonify({'error': 'No file part'}), 400
33     file = request.files['file']
34
35     # Check if the file is empty
36     if file.filename == '':
37         return jsonify({'error': 'No selected file'}), 400
38
39     # Save the uploaded file to the upload folder
40     filename = os.path.join(app.config['UPLOAD_FOLDER'], file.
41                           filename)
42     file.save(filename)
43
44     # Print the saved path of the uploaded image
45     saved_path = os.path.abspath(filename)
46     print('Image saved at:', saved_path)
47     pre_process_result = client.predict(
48         saved_path, # filepath in 'Input Image' Image component
49         True, # bool in 'Remove Background' Checkbox component
50         0.85, # float (numeric value between 0.5 and 1.0) in 'Foreground
51         Ratio' Slider component
52         api_name="/ preprocess"
53     )
54     print( pre_process_result )
55
56     # Perform asynchronous prediction
57     loop = asyncio.new_event_loop()
58     result = loop.run_until_complete(predict_async(pre_process_result
59                                         ))
60     loop.close()
61     print(str(result))
62     new_path = shutil.copy(result, "../scene/bikes/")
63     # Return the result as JSON response
64     print(new_path[3:])
65     return jsonify({'result': new_path[3:]}), 200
66
67 # Asynchronous function to perform prediction
68 async def predict_async(saved_path):
69     # Import gradio_client here to prevent import errors during
70     # synchronous execution
71
72     result = client.predict(
73         gradio_client.file(saved_path),
74         256,
75         api_name="/ generate"
76     )
77     return result[0]
78
79 # Run the Flask app
80 if __name__ == '__main__':

```

```
77     app.run(debug=True)
```

6.3.2 index.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8" />
6   <meta name="viewport" content="width=device-width, initial-scale=1,
7     shrink-to-fit=no" />
8   <meta http-equiv="x-ua-compatible" content="ie=edge" />
9   <title>Custom Bikes Garage</title>
10  <!-- Font Awesome -->
11  <link rel="stylesheet" href="https://use.fontawesome.com/releases/
12    v6.0.0/css/all.css" />
13  <!-- Google Fonts Roboto -->
14  <link rel="stylesheet" href="https://fonts.googleapis.com/css2?
15    family=Roboto:wght@300;400;500;700&display=swap" />
16  <!-- MDB -->
17  <link rel="stylesheet" href="css/mdb.min.css" />
18 <style>
19   *
20     {
21       box-sizing: border-box;
22     }
23
24   html {
25     font-family: Roboto, sans-serif;
26   }
27
28   body {
29     background-color: #080808;
30     color: white;
31   }
32
33   canvas,
34   img,
35   svg {
36     vertical-align: middle;
37   }
38
39   label {
40     cursor: pointer;
41     display: block;
42
43     >input[type="radio"],
44     >input[type="checkbox"] {
45       margin-right: 5px;
46       position: relative;
47       top: -2px;
48     }
49   }
50
51   form {
52     margin-bottom: 20px;
```

```
49      }
50
51  textarea {
52    resize: vertical;
53  }
54
55  svg {
56    vertical-align: middle;
57  }
58
59  .sidebar {
60    background-color: #181718;
61    float: left;
62    height: 100vh;
63    overflow-y: auto;
64    padding: 20px;
65    position: relative;
66    width: 250px;
67    z-index: 1;
68  }
69
70  .main-container {
71    align-items: center;
72    display: flex;
73    height: 100vh;
74    justify-content: center;
75    margin-left: 250px;
76    overflow: auto;
77  }
78
79  .main-container.dark {
80    background-color: #1e1e1e;
81  }
82
83  .field+.field {
84    margin-top: 20px;
85  }
86
87  .field>label {
88    margin-bottom: 5px;
89  }
90
91  .input {
92    background-color: #444;
93    border: 0;
94    border-radius: 4px;
95    color: #fff;
96    display: block;
97    font-size: inherit;
98    outline: 0;
99    padding: 10px 14px;
100   width: 100%;
101   cursor: pointer;
102  }
103 </style >
104
105 </head >
106
```

```

107 <body>
108   <!--Main Navigation-->
109   <header >
110     <style >
111       #intro {
112         background -image: url("/img/Hero\ image.png");
113         height: 100 vh;
114     }
115
116   /* Height for devices larger than 576 px */
117   @media (min-width: 992 px) {
118     #intro {
119       margin -top: -58.59 px;
120     }
121   }
122
123   .navbar .nav -link {
124     color: #fff !important;
125   }
126 </style >
127
128 <!-- Navbar -->
129 <nav class="navbar navbar-expand-lg navbar-dark d-none d-lg-block
130   " style="z-index: 2000;">
131   <div class="container-fluid">
132     <!-- Navbar brand -->
133     <a class="navbar-brand nav-link" target="_blank" href="https
134       ://mdbootstrap.com/docs/standard/">
135       <strong >MDB </ strong >
136     </ a >
137     <button class="navbar-toggler" type="button" data-mdb-
138       collapse-init data-mdb-target="#navbarExample01"
139       aria-controls="navbarExample01" aria-expanded="false" aria-
140       label="Toggle navigation">
141       <i class="fas fa-bars"></ i >
142     </ button >
143     <div class="collapse navbar-collapse" id="navbarExample01">
144       <ul class="navbar-nav me-auto mb-2 mb-lg-0">
145         <li class="nav-item active">
146           <a class="nav-link" aria-current="page" href="#intro">
147             Home </ a >
148         </ li >
149         <li class="nav-item">
150           <a class="nav-link" href="https://www.youtube.com/
151             channel/UC5CF7mLQZhvx8O5GODZAhA"
152             rel="nofollow" target="_blank"></ a >
153         </ li >
154         <li class="nav-item">
155           <a class="nav-link" href="https://mdbootstrap.com/docs/
156             standard/" target="_blank"></ a >
157         </ li >
158       </ ul >
159
160       <ul class="navbar-nav d-flex flex-row">
161         <!-- Icons -->
162         <li class="nav-item me-3 me-lg-0">
163           <a class="nav-link" href="" rel="nofollow" target="
164             _blank">

```

```

157         <i class="fab fa-youtube"></i>
158     </a>
159 </li>
160 <li class="nav-item me-3 me-lg-0">
161     <a class="nav-link" href="https://www.facebook.com/
162         mdbbootstrap" rel="nofollow"
163         target="_blank">
164         <i class="fab fa-facebook-f"></i>
165     </a>
166 </li>
167 <li class="nav-item me-3 me-lg-0">
168     <a class="nav-link" href="https://twitter.com/
169         MDBBootstrap" rel="nofollow" target="_blank">
170         <i class="fab fa-twitter"></i>
171     </a>
172 </li>
173 <li class="nav-item me-3 me-lg-0">
174     <a class="nav-link" href="https://github.com/
175         mdbbootstrap/mdb-ui-kit" rel="nofollow"
176         target="_blank">
177         <i class="fab fa-github"></i>
178     </a>
179 </li>
180 </ul>
181 </div>
182 </div>
183 <!-- Navbar -->
184
185 <!-- Background image -->
186 <div id="intro" class="bg-image shadow-2-strong">
187     <div class="mask" style="background-color: rgba(0, 0, 0, 0.8);"
188         >
189         <div class="container d-flex align-items-center justify-
190             content-center text-center h-100">
191             <div class="text-white" data-mdb-theme="dark">
192                 <h1 class="mb-3">The Garage</h1>
193                 <h5 class="mb-4">Best & free guide of customizing bike
194                     design </h5>
195                 <a class="btn btn-outline-light btn-lg m-2" data-mdb-
196                     ripple-init href="" role="button"
197                     rel="nofollow" target="_blank">Start
198                     Building </a>
199                 <a class="btn btn-outline-light btn-lg m-2" data-mdb-
200                     ripple-init href="" target="_blank"
201                     role="button">About Us</a>
202             </div>
203         </div>
204     </div>
205     <!-- Background image -->
206 </header>
207 <!--Main Navigation-->
208
209 <!--Main layout-->
210 <main class="mt-5">
```

```

207      <div class="container">
208          <!--Section: Content-->
209          <section>
210              <div class="row">
211                  <div class="col-md-6 gx-5 mb-4">
212                      <div class="bg-image hover-overlay shadow-2-strong" data-
213                          mdb-ripple-init
214                          data-mdb-ripple-color="light">
215                          
216                          <a href="#">
217                              <div class="mask" style="background-color: rgba(251,
218                                  251, 251, 0.15);"></div>
219                          </a>
220                      </div>
221                  <div class="col-md-6 gx-5 mb-4">
222                      <h4><strong>Customizing Own Motorcycle</strong></h4>
223                      <p class="text-muted">
224                          Personalizing a motorcycle is a thrilling journey of
225                          creativity and self-expression. For
226                          many riders, their motorcycle is an extension of their
227                          personality, and customizing it
228                          allows them to showcase their unique style and taste.
229                          Whether it's a total overhaul or
230                          subtle tweaks, motorcycle customization opens a world
231                          of possibilities to transform a
232                          factory model into a one-of-a-kind ride. In this
233                          comprehensive guide, we delve into various
234                          motorcycle customization ideas, providing
235                          inspiration and practical tips to make your bike
236                          truly yours.
237                      </p>
238                      <p><strong>The Paint Job</strong></p>
239                      <p class="text-muted">
240                          A custom paint job can dramatically change your
241                          motorcycle's appearance, making it a
242                          powerful way to express your personal style. Options
243                          range from a simple color change to
244                          intricate designs or murals.
245                      </p>
246                  </div>
247              </div>
248          </section>
249          <!--Section: Content-->
250          <section>
251              <div class="row">
252                  <div class="col-md-6 gx-5 mb-4">
253                      <h4><strong>The Customizable Parts</strong></h4>
254                      <p class="text-muted">
255                          Custom seats can drastically improve riding comfort and
256                          the overall look of your motorcycle.
257                          Custom seats can be tailored to your body shape, riding
258                          style, and desired aesthetics.<br>
259                          Wheels can significantly impact a motorcycle's
260                          handling, performance, and appearance.
261                          Options include alloy wheels for a modern sporty look,
262                          wire-spoke wheels for a retro or

```

```
251     classic style, and even custom designs.<br>
252     Accessories offer endless possibilities for
253         personalization. Custom handlebars can change
254         your riding position and bikes look. Mirrors, grips,
255         and levers can be swapped for versions
256         that better match your style. Personalized decals,
257         badges, or tank pads can add a unique
258             touch .
259     </p>
260
261     <p>
262         <h2><strong> The Joy of Customization </strong></h2>
263     </p>
264
265     <p class="text-muted">
266         Theres a certain joy and satisfaction in riding a
267             motorcycle that's been customized
268             to your
269                 liking. Its not just about having a bike that stands
270                     out from the crowd its
271             about the
272                 sense of ownership and connection that comes from
273                     knowing your motorcycle inside and
274                         out.<br>
275
276     In conclusion, motorcycle customization is a journey of
277         creativity, learning, and
278             self-expression. Whether you're a seasoned rider
279                 looking to add a personal touch to
280                     your
281                         ride or a newbie wanting to make your first bike truly
282                             your own, the world of
283                                 motorcycle
284                                     customization is full of possibilities waiting to be
285                                         explored.
286     </p>
287 </div >
288 <div class="col-md-6 gx-5 mb-4">
289     <div class="bg-image hover-overlay shadow-2-strong" data-
290         mdb-ripple-init
291         data-mdb-ripple-color="light">
292             
294             <a href="#">
295                 <div class="mask" style="background-color: rgba(251,
296                     251, 251, 0.15);"></div>
297             </a>
298         </div >
299     </div >
300 </div >
301 </section >
302
303     <hr class="my-5" />
304 <div class="sidebar">
305     <form>
306         <div class="field">
307             <input type="button" name="color" class="input" value="
```

```

296     Full Screen" id="center-view">
297   </div>
298   <div class="field">
299     <input type="button" name="color" class="input" value="
300       Change Bike" id="first-feature">
301   </div>
302
303   <div class="field">
304     <input type="color" name="color" class="input" value="
305       Change Color" id="colorPicker">
306   </div>
307
308   <div class="field">
309     <input type="button" name="color" class="input" value="
310       Change Color" id="second-feature">
311   </div>
312
313   <div class="field">
314     <input type="file" accept="image/*" name="color" class="input"
315       id="bike-image">
316   </div>
317
318   <div class="field">
319     <input type="button" name="color" class="input" value="
320       Upload Bike" id="upload-bike">
321   </div>
322
323   </form>
324 </div>
325 <div class="main-container dark" id="container">
326   </div>
327
328   <hr class="my-5" />
329   <!--Section: Content-->
330   <section class="text-center">
331     <h4 class="mb-5"><strong></strong></h4>
332
333   <div class="row">
334     <div class="col-lg-4 col-md-12 mb-4">
335       <div class="card">
336         <div class="bg-image hover-overlay" data-mdb-ripple-
337           init data-mdb-ripple-color="light">
338           
339           <a href="#">
340             <div class="mask" style="background-color: rgba(
341               251, 251, 251, 0.15);"></div>
342           </a>
343         </div>
344       </div>
345     </div>
346
347   <div class="col-lg-4 col-md-6 mb-4">
348     <div class="card">
349       <div class="bg-image hover-overlay" data-mdb-ripple-
350           init data-mdb-ripple-color="light">
351           
352           <a href="#">
353             <div class="mask" style="background-color: rgba(
354               251, 251, 251, 0.15);"></div>

```

```
        (251, 251, 251, 0.15);"></div>
345      </a>
346    </div >
347  </div >
348</div >
349
350<div class="col-lg-4 col-md-6 mb-4">
351  <div class="card">
352    <div class="bg-image hover-overlay" data-mdb-ripple-
353      init data-mdb-ripple-color="light">
354        
355        <a href="#">
356          <div class="mask" style="background-color: rgba
357            (251, 251, 251, 0.15);"></div>
358        </a>
359      </div >
360    </div >
361  <div class="col-lg-4 col-md-6 mb-4">
362    <div class="card">
363      <div class="bg-image hover-overlay" data-mdb-ripple-
364        init data-mdb-ripple-color="light">
365          
366          <a href="#">
367            <div class="mask" style="background-color: rgba
368              (251, 251, 251, 0.15);"></div>
369          </a>
370        </div >
371      </div >
372    <div class="col-lg-4 col-md-6 mb-4">
373      <div class="card">
374        <div class="bg-image hover-overlay" data-mdb-ripple-
375          init data-mdb-ripple-color="light">
376            
377            <a href="#">
378              <div class="mask" style="background-color: rgba
379                (251, 251, 251, 0.15);"></div>
380            </a>
381          </div >
382        <div class="col-lg-4 col-md-6 mb-4">
383          <div class="card">
384            <div class="bg-image hover-overlay" data-mdb-ripple-
385              init data-mdb-ripple-color="light">
386                
387                <a href="#">
388                  <div class="mask" style="background-color: rgba
389                    (251, 251, 251, 0.15);"></div>
390                </a>
391              </div >
392            </div >
393          </section >
394          <!--Section: Content-->
```

```
394 <hr class="my-5" />
395
396 <!--Section: Content-->
397 <section class="mb-5">
398   <h4 class="mb-5 text-center"><strong>For more info contact us
      </strong></h4>
399
400   <div class="row d-flex justify-content-center">
401     <div class="col-md-6">
402       <form>
403         <!-- 2 column grid layout with text inputs for the
             first and last names -->
404         <div class="row mb-4">
405           <div class="col">
406             <div class="form-outline" data-mdb-input-init>
407               <input type="text" id="form3Example1" class="form-control" />
408               <label class="form-label" for="form3Example1">
409                 First name</label>
410             </div>
411           </div>
412           <div class="col">
413             <div class="form-outline" data-mdb-input-init>
414               <input type="text" id="form3Example2" class="form-control" />
415               <label class="form-label" for="form3Example2">
416                 Last name</label>
417             </div>
418           </div>
419         </div>
420
421         <!-- Email input -->
422         <div class="form-outline mb-4" data-mdb-input-init>
423           <input type="email" id="form3Example3" class="form-control" />
424           <label class="form-label" for="form3Example3">Email
425             address</label>
426         </div>
427
428         <!-- Password input -->
429         <div class="form-outline mb-4" data-mdb-input-init>
430           <input type="password" id="form3Example4" class="form-control" />
431           <label class="form-label" for="form3Example4">
432             Password</label>
433         </div>
434
435         <!-- Checkbox -->
436         <div class="form-check d-flex justify-content-center mb-4">
437           <input class="form-check-input me-2" type="checkbox"
438             value="" id="form2Example3"
439             checked />
440           <label class="form-check-label" for="form2Example3">
441             Subscribe to our newsletter
442           </label>
443         </div>
444
```

```

440      <!-- Submit button -->
441      <button type="submit" class="btn btn-primary btn-block
        mb-4" data-mdb-ripple-init>
442          Sign up
443      </button >
444
445      <!-- Register buttons -->
446      <div class="text-center">
447          <p>or sign up with:</p>
448          <button type="button" class="btn btn-primary btn-
            floating mx-1" data-mdb-ripple-init>
449              <i class="fab fa-facebook-f"></i>
450          </button >
451
452          <button type="button" class="btn btn-primary btn-
            floating mx-1" data-mdb-ripple-init>
453              <i class="fab fa-google"></i>
454          </button >
455
456          <button type="button" class="btn btn-primary btn-
            floating mx-1" data-mdb-ripple-init>
457              <i class="fab fa-twitter"></i>
458          </button >
459
460          <button type="button" class="btn btn-primary btn-
            floating mx-1" data-mdb-ripple-init>
461              <i class="fab fa-github"></i>
462          </button >
463      </div >
464      </form >
465  </div >
466  </div >
467  </section >
468  <!--Section: Content-->
469  </div >
470  </main >
471  <!--Main layout-->
472
473  <script type="module" src="/main.js"></script>
474  <script type="module" src="/app.js"></script>
475 </body >
476
477 </html >
```

6.3.3 main.js

```

1 import * as THREE from 'three';
2 import { GLTFLoader } from 'three/addons/loaders/GLTFLoader.js';
3 import { OBJLoader } from 'three/addons/loaders/OBJLoader.js';
4 import { OrbitControls } from 'three/addons/controls/OrbitControls.js
  ;
5 import { client } from "@gradio/client";
6
7 const scene = new THREE.Scene();
8 scene.background = new THREE.Color(0xffffffff);
```

```

9 const camera = new THREE.PerspectiveCamera(75, window.innerWidth /
    window.innerHeight, 0.1, 1000);
10 const loader = new GLTFLoader();
11 const obj_loader = new OBJLoader();
12
13 const renderer = new THREE.WebGLRenderer();
14 renderer.setSize(window.innerWidth, window.innerHeight);
15 const parentNode = document.getElementById("container");
16
17 parentNode.appendChild(renderer.domElement);
18 const controls = new OrbitControls(camera, renderer.domElement);
19
20
21
22 const bikeData = await loader.loadAsync('scene/bikes/2020
    _suzuki_sv650_motorcycle.glb');
23 const parkingData = await loader.loadAsync('scene/parking_garage.glb
    ');
24 const workbenchData = await loader.loadAsync('scene/old_work_bench.
    glb');
25
26
27 camera.position.z = 5;
28 scene.add(parkingData.scene.children[0]);
29 scene.add(bikeData.scene.children[0]);
30
31 await loader.load(
32     'scene/parking_garage.glb',
33     function (gltf) {
34         scene.add(gltf.scene);
35     });
36
37 // await loader.load(
38 //     'scene/2020_suzuki_sv650_motorcycle.glb',
39 //     function (gltf) {
40 //         scene.add(gltf.scene);
41 //     });
42
43 // console.log(workbenchData.scene.children[0]);
44
45 let moveMouse = new THREE.Vector2();
46
47 window.addEventListener("mousemove", (event) => {
48     moveMouse.x = (event.clientX / window.innerWidth) * 2 - 1;
49     moveMouse.y = -(event.clientY / window.innerHeight) * 2 + 1;
50
51     camera.position.x = moveMouse.x;
52     camera.position.y = moveMouse.y;
53     controls.update();
54
55     renderer.render(scene, camera);
56 });
57 renderer.render(scene, camera);
58
59 const centerViewButton = document.getElementById("center-view");
60 /* Get the documentElement (<html>) to display the page in fullscreen
   */
61 var elem = document.body;

```

```

62
63 /* View in fullscreen */
64 function openFullscreen() {
65   if (elem.requestFullscreen) {
66     elem.requestFullscreen();
67   } else if (elem.webkitRequestFullscreen) { /* Safari */
68     elem.webkitRequestFullscreen();
69   } else if (elem.msRequestFullscreen) { /* IE11 */
70     elem.msRequestFullscreen();
71   }
72 }
73 centerViewButton.addEventListener("click", (event) => {
74   openFullscreen();
75 })
76 const ambientLight = new THREE.AmbientLight(0xffffff, 0.4);
77 scene.add(ambientLight);
78
79 const dirLight = new THREE.DirectionalLight(0xefefff, 1.5);
80 dirLight.position.set(10, 10, 10);
81 scene.add(dirLight);
82
83 function* getBikesModel() {
84   yield "scene/bikes/kawasaki_ninja_h2r.glb";
85   yield "scene/bikes/three_cylinder_naked_street_bike.glb";
86 }
87
88 // Calling the Generate Function
89 let gen = getBikesModel();
90
91 const firstFeatureButton = document.getElementById("first-feature");
92 firstFeatureButton.addEventListener("click", async (event) => {
93   var bikeChild = scene.children[1];
94   scene.remove(bikeChild);
95   await loader.load(
96     gen.next().value,
97     function (gltf) {
98       scene.add(gltf.scene);
99       gltf.scene.position.set(1, -0.5, 0);
100    });
101 })
102
103
104 let selectedColor = new THREE.Color(0xffffff);
105 function handleColorChange(event) {
106   selectedColor.set(event.target.value);
107 }
108 const colorPicker = document.getElementById('colorPicker');
109 colorPicker.addEventListener('change', handleColorChange);
110
111 renderer.domElement.addEventListener('click', (event) => {
112   var mouse = {
113     x: (event.clientX / window.innerWidth) * 2 - 1,
114     y: -(event.clientY / window.innerHeight) * 2 + 1,
115   };
116   var objChild = scene.children[1];
117   var raycaster = new THREE.Raycaster();
118   raycaster.setFromCamera(mouse, camera);
119   var intersects = raycaster.intersectObjects(scene.children, true);

```

```

120   console.log(intersects)
121   if (intersects.length > 0) {
122     const clickedObject = intersects[0].object;
123     if (clickedObject.material) {
124       clickedObject.material.color.copy(selectedColor);
125       console.log("color should change")
126     }
127   }
128 });
129
130 function* getDifferentColorBikesModel() {
131   yield "scene/bikes/kawasaki_ninja_h2r.glb";
132   yield "scene/bikes/scene123";
133 }
134
135
136 // Calling the Generate Function
137 let gdcb = getDifferentColorBikesModel();
138
139 const secondFeatureButton = document.getElementById("second-feature")
  ;
140 secondFeatureButton.addEventListener("click", async (event) => {
141   var bikeChild = scene.children[1];
142   scene.remove(bikeChild);
143   await loader.load(
144     gdcb.next().value,
145     function (gltf) {
146       scene.add(gltf.scene);
147       gltf.scene.position.set(1, -0.5, 0);
148     });
149 })
150
151 // code for handling upload new bike button
152 const uploadBikeButton = document.getElementById("upload-bike");
153 uploadBikeButton.addEventListener('click', async (event) => {
154   var fileInput = document.getElementById('bike-image');
155   var file = fileInput.files[0];
156   var formData = new FormData();
157   formData.append('file', file);
158
159   // Send a POST request to the /upload endpoint
160   await fetch('http://127.0.0.1:5000/upload', {
161     method: 'POST',
162     body: formData
163   })
164   .then(response => response.json())
165   .then(async data => {
166     // console.log('Saved path:', data.result);
167     // alert('Image uploaded successfully! Saved path: ' + data.
168     result);
169     var bikeChild = scene.children[1];
170     scene.remove(bikeChild);
171     await obj_loader.load(
172       data.result,
173       function (gltf) {
174         scene.add(gltf);
175       });
176     });
177   }

```

```
176     .catch(error => {
177       console.error('Error uploading image:', error);
178       alert('Error uploading image: ' + error.message);
179     });
180 })
```