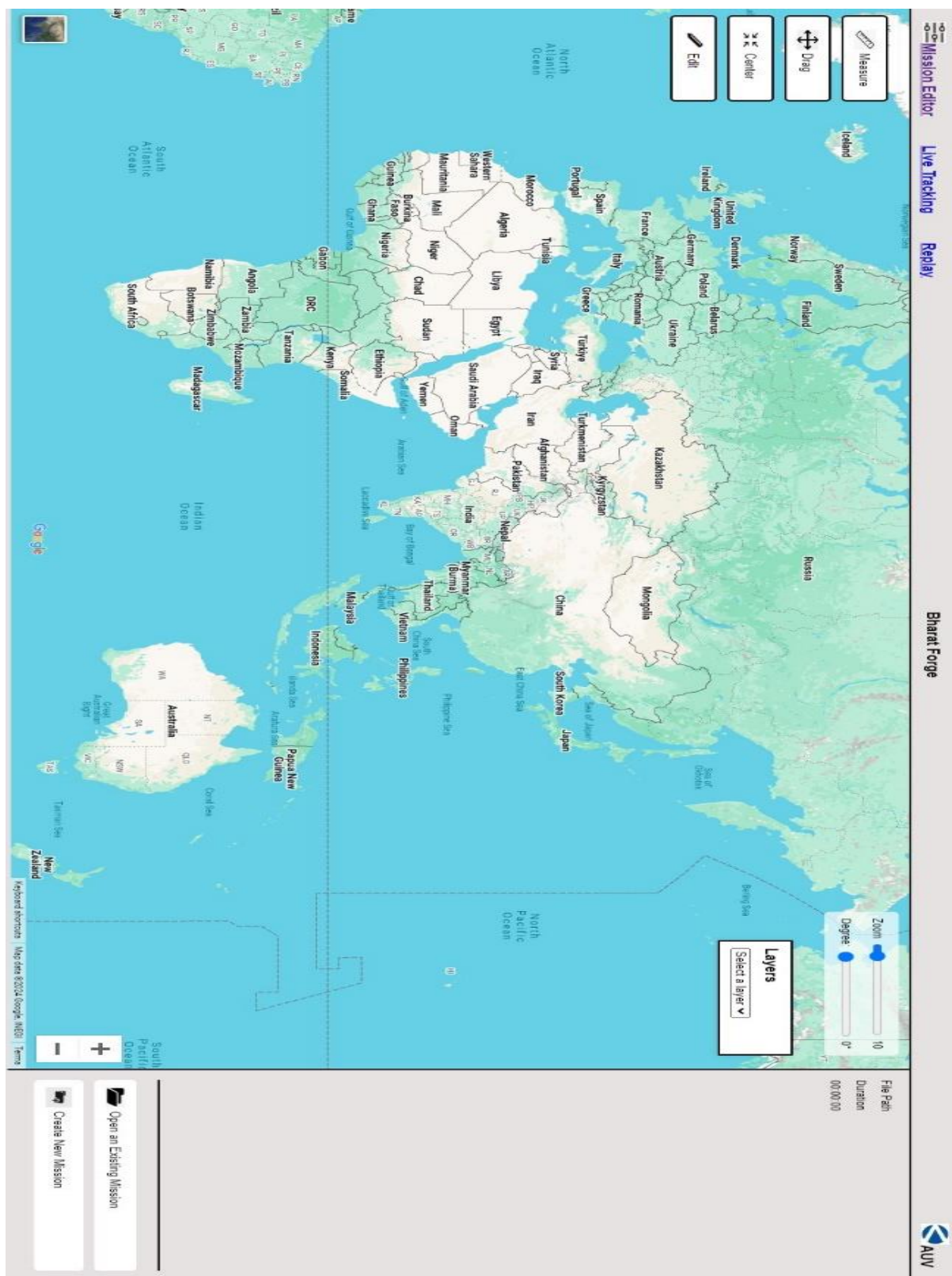


FRONT INTERFACE:



General

Theme
Light ▼

Language
English ▼

Sound Off

Mission Default Values

WP Speed (m/s)
0
Increase WP Speed

WP Altitude (m)
0
Increase WP Altitude

Rails Speed (m/s)
0
Increase Rails Speed

Rails Altitude (m)
0
Increase Rails Altitude

Track Family Interval (ms)
0
Increase Track Family Interval

Circle Speed (m/s)
0
Increase Circle Speed

Circle Altitude (m)
0
Increase Circle Altitude

Timeout (s)
0
Increase Timeout

Units

Distance (m)
m ▼

Coordinate
D ▼

Speed (m/s)
m/s ▼

Area Default Values

Exclusion
Border ☐ Fill ☐

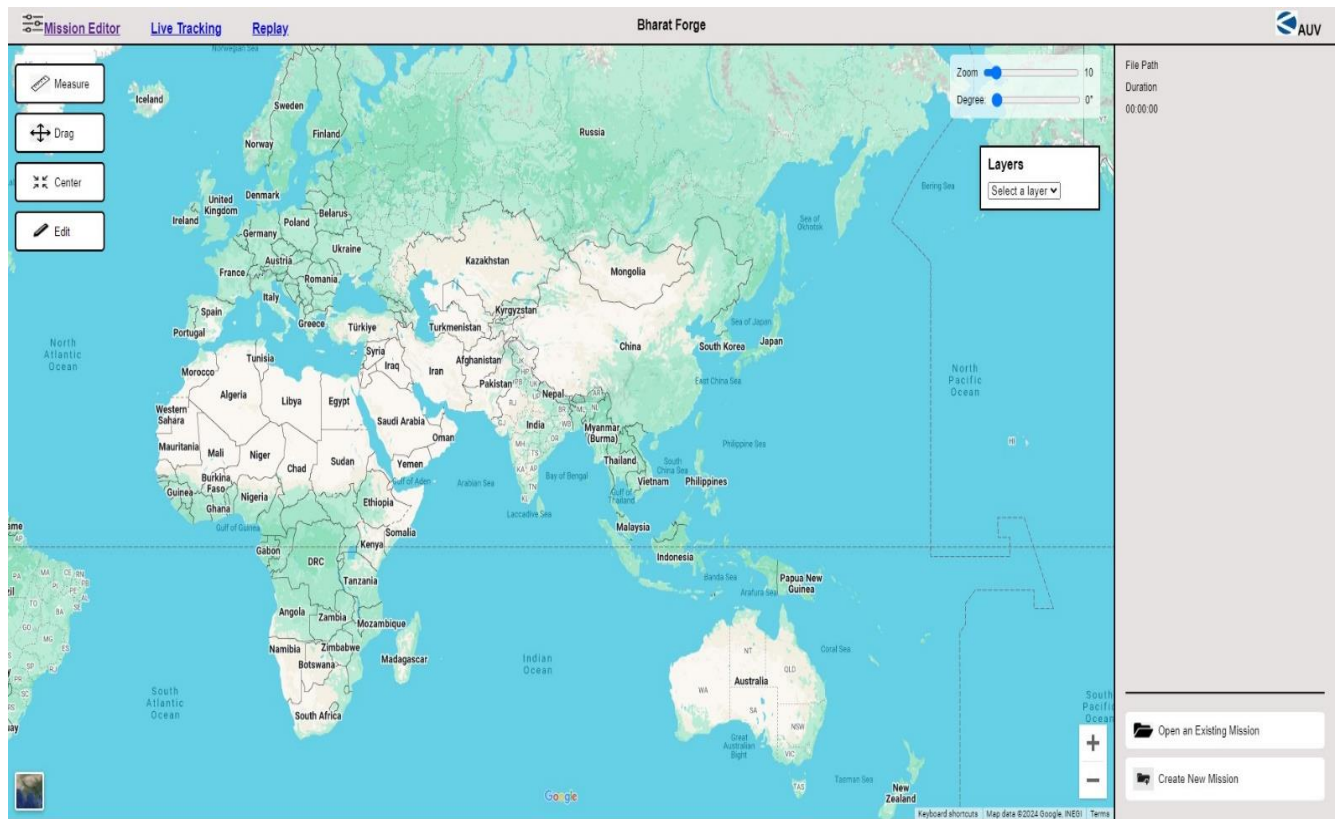
Containment
Border ☐ Fill ☒

Min Depth
Border ☐ Fill ☒

Max Depth
Border ☐ Fill ☒

Contacts
☐

FRONT INTERFACE :



HTML CODE OF FRONT INTERFACE :

HTML CODE :

[illegible]

```
Tracking</a>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<a  
href="https://www.live_tracking.com">Replay</a></b></div>  
    <div class="center">  
        <b>Bharat Forge</b>  
    </div>  
    <div class="right"> <b>AUV</b></div>  
</div>  
<div class="divider1"></div>  
<div class="main">  
    <div class="center-panel">  
        <iframe  
src="https://www.google.com/maps/embed?pb=!1m14!1m12!1m3!1d49681293.333823  
5!2d85.9709283465221!3d21.51929459709748!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!  
4f13.1!5e0!3m2!1sen!2sin!4v1725011614635!5m2!1sen!2sin" width="1600"  
height="850" style="border:0;" allowfullscreen="" loading="lazy"  
referrerpolicy="no-referrer-when-downgrade"></iframe>  
  
        <!-- Transparent Buttons on the Left -->  
        <div class="transparent-buttons">  
            <button>  
                Measure  
            </button>  
            <button>  
                Drag  
            </button>  
            <button>  
                Center  
            </button>  
            <button>  
                Edit  
            </button>  
        </div>  
  
        <!-- Transparent Controls on the Right (Zoom and Degree) -->  
        <div class="transparent-controls">  
            <div class="zoom-panel">  
                <label for="zoomRange">Zoom</label>  
                <input type="range" id="zoomRange" min="1" max="100"  
value="10" onchange="updateZoomValue()">  
                <span id="zoomValue">10</span>  
            </div>  
            <div class="degree-display">  
                <label for="degreeRange">Degree:</label>
```

```

        <input type="range" id="degreeRange" min="0" max="360"
value="0" onchange="updateDegreeValue()">
        <span id="degreeValue">0°</span>
    </div>
</div>

<!-- Layer Box on the Right -->
<div class="layer-box">
    <div class="layer-header">Layers</div>
    <select id="layerSelect" onchange="addLayer()">
        <option value="" disabled selected>Select a
layer</option>
        <option value="S57">S57</option>
        <option value="S63">S63</option>
        <option value="GeoTIFF">GeoTIFF</option>
        <option value="WMTS">WMTS</option>
        <option value="WMS">WMS</option>
    </select>
    <div class="layer-list">
        <!-- Dynamically added layers will appear here -->
    </div>
</div>
</div>
<div class="divider2"></div>
<div class="right-panel">
    <div class="file-info">
        <div>File Path</div>
        <div>Duration</div>
        <div id="durationDisplay">00:00:00</div>
    </div>

    <div class="divider"></div>

    <div class="mission-buttons">
        <button>
            
            Open an Existing Mission
        </button>
        <button>
            
            Create New Mission
        </button>
    </div>
</div>
</div>

```

```

<script>
  function updateZoomValue() {
    const zoomRange = document.getElementById('zoomRange');
    const zoomValue = document.getElementById('zoomValue');
    zoomValue.textContent = zoomRange.value;
    // You can add functionality here to adjust the map zoom if
needed
  }

  function updateDegreeValue() {
    const degreeRange = document.getElementById('degreeRange');
    const degreeValue = document.getElementById('degreeValue');
    degreeValue.textContent = `${degreeRange.value}°`;
    // You can add functionality here to rotate the map if needed
  }

  function addLayer() {
    const select = document.getElementById('layerSelect');
    const layerName = select.value;
    const layerList = document.querySelector('.layer-list');
    const existingLayer =
Array.from(layerList.children).find(layer =>
layer.textContent.includes(layerName));

    if (!existingLayer) {
      const layerItem = document.createElement('div');
      layerItem.className = 'layer-item';
      layerItem.innerHTML = `${layerName} <button
onclick="removeLayer(this)"></button>;
      layerList.appendChild(layerItem);
    }
  }

  function removeLayer(button) {
    const layerItem = button.parentElement;
    layerItem.remove();
  }
</script>

</body>
</html>

```

EXPLANATION OF HTML CODE :

This code creates a simple web-based mission editor interface using HTML, CSS, and JavaScript. Below is a breakdown of the key components:

1. HTML Structure

- <head> Section:

- The head section includes metadata about the webpage such as character encoding and viewport settings for responsive design.

- The stylesheet final mission editor.css is linked for custom styling.

- <body> Section:

- Header (<div class="header">):

- Contains three main sections:

- **Left:** Displays a "Mission Editor" link, along with links to "Live Tracking" and "Replay".

- **Center:** Displays the text "Bharat Forge".

- **Right:** Displays an image (logo) with the text "AUV".

- Main Content (<div class="main">):

- Center Panel:

- Google Maps Embed: Displays an embedded Google Map using an <iframe>.

- Transparent Buttons: Positioned on the left, these buttons allow users to interact with the map (e.g., Measure, Drag, Center, Edit).

- Transparent Controls: Positioned on the right, this section includes controls for zooming and rotating (degree) the map.

- Layer Box: Also on the right, this box allows users to select and manage different map layers (S57, S63, GeoTIFF, WMTS, WMS).

- Right Panel (<div class="right-panel">):

- File Info: Displays file-related information like "File Path" and "Duration".

- Mission Buttons: Provides options to "Open an Existing Mission" or "Create New Mission".

2. JavaScript Functionality

- Update Zoom Value (updateZoomValue):
 - Retrieves the current value from the zoom slider and updates the displayed zoom value.
- Update Degree Value (updateDegreeValue):
 - Retrieves the current value from the degree slider and updates the displayed degree value.
- Add Layer (addLayer):
 - Adds a new layer to the "Layer Box" when the user selects a layer from the dropdown. If the layer is already present, it does not add it again.
- Remove Layer (removeLayer):
 - Removes a layer from the "Layer Box" when the corresponding remove button is clicked.

3. User Interface Layout

- The layout is designed to be visually clean, with a header, a main content area (for map and layers), and a right panel for additional mission options.
- Controls and buttons are designed to be intuitive and directly related to map interactions, making the UI user-friendly for editing and managing missions.

4. CSS and Image Dependencies

- External CSS (final mission editor.css) is used for styling, though it's not included in the provided code.
- Various images (e.g., setting.png, Kalyani1.jpg, etc.) are used for icons and visuals in the header and buttons. These images are assumed to be in the same directory as the HTML file.

5. Customization and Extensibility

- The map controls and layer management system can be extended with additional features or improved map interactions.
- The UI structure can be customized further by modifying the CSS file.

CSS CODE OF FRONT INTERFACE :

```
body{
  margin: 0;
  font-family: Arial, Helvetica, sans-serif;
  background-color: rgb(230, 226, 226);
}
.header{
  display: flex;
  justify-content: space-between;
  align-items: center;
  padding: 5px 20px;
  background-color: rgb(230, 226, 226);
  color:black;
  font-size: medium;
}

.header .left,
.header .center,
.header .right{
  flex: 1;
}
.header .center {
  text-align: center;
}

.header .right {
  text-align: right;
}
.divider1{
  border-bottom: 2px solid black;
  margin: 0 0;
}
.divider2{
  border-left: 2px solid black;
  margin: 0 0;
}

.main{
```

```

    display: flex;
    height: calc(100vh - 100px);
}
.center-panel{
    flex: 1;
    display: flex;
    justify-content: flex-start;
    position: relative;
}
.transparent-buttons {
    position: absolute;
    top: 20px;
    left: 10px;
    display: flex;
    flex-direction: column;
    gap: 10px;
}

.transparent-buttons button {
    background-color: white;
    border: 2px solid 000;
    padding: 10px 20px;
    border-radius: 5px;
    color: black;
    cursor: pointer;
    font-size: 13px;
    text-align: center;
    backdrop-filter: blur(5px);
    display: flex;
    align-items: center;
    gap: 5px;
}

.transparent-buttons button:hover {
    background-color: rgba(255, 255, 255, 0.7);
}

/ New Transparent Controls /
.transparent-controls {
    position: absolute;
    top: 10px; / Adjust this value to position above the layer box /
    right: 20px;
    background-color: rgba(255, 255, 255, 0.7);
    padding: 10px;
    border-radius: 5px;
    z-index: 1000;
    font-size: 12px;
    color: black;

```

```

display: flex;
flex-direction: column;
gap: 10px;
}

.transparent-controls div {
display: flex;
align-items: center;
gap: 5px;
}

.transparent-controls input[type="range"] {
width: 100%;
}

.layer-box {
position: absolute;
top: 110px; / Adjusted to make space for transparent-controls /
right: 20px;
width: 150px;
border: 2px solid black;
background-color: white;
padding: 10px;
box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1);
z-index: 999;
}

.layer-header {
font-weight: bold;
margin-bottom: 10px;
}

.layer-list {
font-size: 12px;
}

.right-panel{
width: 20%;
display: flex;
flex-direction: column;
justify-content: space-between;
background-color:rgb(230, 226, 226);
padding: 15px;
}

.file-info{
text-align: left;
margin-bottom: auto;
}

.file-info div{

```

```

margin-bottom: 10px;
font-size: 12px;
color: black;
}
.divider{
border-bottom: 2px solid black;
margin: 20px 0;
}
.mission-buttons {
text-align: right;
}

.mission-buttons button{
width: 100%;
padding: 10px;
background-color: white;
color:black;
border: none;
border-radius: 5px;
cursor: pointer;
font-size: 13px;
margin-bottom: 10px;
text-align: center;
display: flex;
align-items: center;
gap: 5px;
}
.durationDisplay{
font-size: 12px;
}

```

EXPLANATION OF CSS CODE :

This CSS code defines the styles for a web-based mission editor interface. Here's a breakdown of the key components and their roles:

1. General Styles

- body:
 - Sets the margin to 0 to remove default spacing around the page.
 - Applies a global font (Arial, Helvetica, sans-serif) and sets a background color of rgb(230, 226, 226) to give the entire page a light gray appearance.

2. Header Styles

- .header:

- Uses display: flex to arrange child elements (left, center, right) horizontally.
- Centers the content vertically using align-items: center and spaces them horizontally with justify-content: space-between.
- Padding around the header is set to 5px 20px.
- Background and text color are set to maintain consistency with the overall theme.
- .header .left, .header .center, .header .right:
 - Each of these classes takes up one-third of the header's width (flex: 1).
 - Text alignment is customized: left for the .left section, center for the .center, and right for the .right.

3. Dividers

.divider1 and .divider2:

.divider1: Creates a horizontal divider (border-bottom) below the header.

.divider2: Creates a vertical divider (border-left) within the main content area.

4. Main Content Area

.main:

- Uses display: flex to create a horizontal layout for the center and right panels.
- The height of this section is set to take up the full height of the viewport minus 100 pixels (calc(100vh - 100px)), accounting for the height of the header and dividers.
- .center-panel:
 - Takes up the remaining space after the right panel (flex: 1).
 - Positioned elements inside the center panel are set to start from the top-left (justify-content: flex-start).
 - position: relative allows child elements like the transparent buttons and controls to be absolutely positioned within this panel.

5. Transparent Buttons (Left Side)

- .transparent-buttons:
 - Positioned absolutely within the .center-panel, starting 20 pixels from the top and 10 pixels from the left.

- Buttons are stacked vertically using flex-direction: column with a 10px gap.
- .transparent-buttons button:
 - Styled with a white background, black border, and black text.
 - Buttons have padding, rounded corners (border-radius: 5px), and are designed to be interactive (cursor: pointer).
 - backdrop-filter: blur(5px) adds a subtle blur effect to the background behind the buttons.
 - Hover Effect: Changes the background color to a slightly transparent white when hovered.

6. Transparent Controls (Right Side)

- .transparent-controls:
 - Positioned above the layer box (top: 10px), 20 pixels from the right.
 - Background is semi-transparent (rgba(255, 255, 255, 0.7)) with padding and rounded corners for a subtle look.
 - Elements are stacked vertically with a 10px gap.
- .transparent-controls div:
 - Each control (e.g., zoom, degree) is a flex container with aligned items and a small gap for spacing.
- .transparent-controls input[type="range"]:
 - The range input (slider) spans the full width of its container.

7. Layer Box

- .layer-box:
 - Positioned absolutely within the .center-panel, 110 pixels from the top and 20 pixels from the right.
 - Styled with a white background, black border, padding, and a subtle shadow for depth.
- .layer-header:
 - The header for the layer box is bolded and separated from the rest of the content with a margin.
- .layer-list:

- Displays the list of added layers with smaller font size.

8. Right Panel

- .right-panel:
 - Positioned on the right side of the main content, taking up 20% of the width.
 - Flexbox layout is used for vertical alignment of content, with space between different sections.
 - Background color matches the overall theme.
- .file-info:
 - Aligns text to the left and uses a smaller font size with a black color for clarity.
 - Provides some spacing between elements within this section.
- .divider:
 - Creates a horizontal divider within the right panel to separate different sections.
- .mission-buttons:
 - Buttons are full-width with white backgrounds and black text, rounded corners, and a similar design to the transparent buttons.

9. Miscellaneous

- .durationDisplay:
 - Sets the font size for the duration display in the right panel to be slightly smaller.

SETTING PANEL :

General

Theme
Light

Language
English

Sound: Off

Mission Default Values

WP Speed (m/s) 0 Increase WP Speed

WP Altitude (m) 0 Increase WP Altitude

Rails Speed (m/s) 0 Increase Rails Speed

Rails Altitude (m) 0 Increase Rails Altitude

Track Family Interval (m/s) 0 Increase Track Family Interval

Circle Speed (m/s) 0 Increase Circle Speed

Circle Altitude (m) 0 Increase Circle Altitude

Timeout (s) 0 Increase Timeout

Units

Distance (m) m

Coordinate D

Speed (m/s) m/s

Area Default Values

Exclusion Border ☐ Fill ☐

Containment Border ☐ Fill ☐

Min Depth Border ☐ Fill ☐

Max Depth Border ☐ Fill ☐

Contacts ☐

HTML CODE OF SETTING PANEL:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Setting</title>
  <link rel="stylesheet" href="general setting.css">
</head>
<body>
  <div class="left-panel panel">
    <h4>General</h4>
    <div class="form-group">
      <label for="theme">Theme</label>
```



```

        <select id="theme">
            <option value="light">Light</option>
            <option value="dark">Dark</option>
            <option value="solarized">Solarized</option>a
        </select>
    </div>
    <div class="form-group">
        <label for="language">Language</label>
        <select id="language">
            <option value="en">English</option>
            <option value="es">Spanish</option>
            <option value="fr">French</option>
            <option value="de">German</option>
        </select>
    </div>
    <div class="form-group">
        <button id="sound">Sound: Off</button>
    </div>

    <div class="mission-section">
        <h4>Mission Default Values</h4>
        <div class="mission-item">
            <label for="wp-speed">WP Speed (m/s)</label>
            <input type="number" id="wp-speed" value="0" min="0"
max="300">
            <button onclick="increaseValue('wp-speed')">Increase WP
Speed</button>
        </div>
        <div class="mission-item">
            <label for="wp-altitude">WP Altitude (m)</label>
            <input type="number" id="wp-altitude" value="0" min="0"
max="300">
            <button onclick="increaseValue('wp-altitude')">Increase WP
Altitude</button>
        </div>
        <div class="mission-item">
            <label for="rails-speed">Rails Speed (m/s)</label>
            <input type="number" id="rails-speed" value="0" min="0"
max="300">
            <button onclick="increaseValue('rails-speed')">Increase
Rails Speed</button>
        </div>
        <div class="mission-item">
            <label for="rails-altitude">Rails Altitude (m)</label>
            <input type="number" id="rails-altitude" value="0" min="0"
max="300">
            <button onclick="increaseValue('rails-altitude')">Increase
Rails Altitude</button>

```

```

        </div>
        <div class="mission-item">
            <label for="track-family-interval">Track Family Interval
(m/s)</label>
            <input type="number" id="track-family-interval" value="0"
min="0" max="300">
            <button onclick="increaseValue('track-family-
interval')">Increase Track Family Interval</button>
        </div>
        <div class="mission-item">
            <label for="circle-speed">Circle Speed (m/s)</label>
            <input type="number" id="circle-speed" value="0" min="0"
max="300">
            <button onclick="increaseValue('circle-speed')">Increase
Circle Speed</button>
        </div>
        <div class="mission-item">
            <label for="circle-altitude">Circle Altitude (m)</label>
            <input type="number" id="circle-altitude" value="0"
min="0" max="300">
            <button onclick="increaseValue('circle-
altitude')">Increase Circle Altitude</button>
        </div>
        <div class="mission-item">
            <label for="timeout">Timeout (s)</label>
            <input type="number" id="timeout" value="0" min="0"
max="300">
            <button onclick="increaseValue('timeout')">Increase
Timeout</button>
        </div>
    </div>
</div>
<div class="right-panel panel">
    <h4>Units</h4>
    <div class="unit-item">
        <label for="distance">Distance (m)</label>
        <select id="distance">
            <option value="en">m</option>
            <option value="es">feet</option>
        </select>
    </div>
    <div class="unit-item">
        <label for="coordinate">Coordinate</label>
        <select id="coordinate">
            <option value="en">D</option>
            <option value="es">D</option>
            <option value="es">DWH</option>
            <option value="es">DM</option>

```

```

        <option value="es">DMWH</option>
        <option value="es">DMS</option>
        <option value="es">DMSWH</option>
    </select>
</div>
<div class="unit-item">
    <label for="speed">Speed (m/s)</label>
    <select id="speed">
        <option value="en">m/s</option>
        <option value="es">knots</option>
    </select>
</div>

<h4>Area Default Values</h4>
<div class="area-item">
    <label for="exclusion">Exclusion</label>
    <label for="exclusion">Border</label>
    <div class="circle-button" onclick="toggleButton(this)"></div>
    <label for="exclusion">Fill</label>
    <div class="circle-button" onclick="toggleButton(this)"></div>
</div>
<div class="area-item">
    <label for="containment">Containment</label>
    <label for="containment">Border</label>
    <div class="circle-button" onclick="toggleButton(this)"></div>
    <label for="containment">Fill</label>
    <div class="circle-button" onclick="toggleButton(this)"></div>
</div>
<div class="area-item">
    <label for="min-depth">Min Depth</label>
    <label for="min-depth">Border</label>
    <div class="circle-button" onclick="toggleButton(this)"></div>
    <label for="min-depth">Fill</label>
    <div class="circle-button" onclick="toggleButton(this)"></div>
</div>
<div class="area-item">
    <label for="max-depth">Max Depth</label>
    <label for="max-depth">Border</label>
    <div class="circle-button" onclick="toggleButton(this)"></div>
    <label for="max-depth">Fill</label>
    <div class="circle-button" onclick="toggleButton(this)"></div>
</div>
<div class="area-item">
    <label for="contacts">Contacts</label>
    <div class="circle-button" onclick="toggleButton(this)"></div>
</div>
</div>

```

```

<script>
    // JavaScript to handle the circle button toggle
    function toggleButton(button) {
        button.classList.toggle('active');
    }
</script>
</div>

<script>
    // JavaScript to handle unit updates can go here
</script>
</div>

<script>
    document.getElementById('theme').addEventListener('change',
function() {
    var selectedTheme = this.value;

    // Remove any existing theme classes
    document.body.classList.remove('dark-theme', 'solarized-
theme');

    // Add the selected theme class
    if (selectedTheme === 'dark') {
        document.body.classList.add('dark-theme');
    } else if (selectedTheme === 'solarized') {
        document.body.classList.add('solarized-theme');
    } else {
        // Default to light theme
        document.body.classList.remove('dark-theme', 'solarized-
theme');
    }
    });

    // JavaScript to handle the sound button text
    document.getElementById('sound').addEventListener('click',
function() {
    var button = this;
    if (button.textContent === 'Sound: Off') {
        button.textContent = 'Sound: On';
        // Add logic here to enable sound if needed
    } else {
        button.textContent = 'Sound: Off';
        // Add logic here to disable sound if needed
    }
    });

    // JavaScript to increase values for mission default parameters

```

```

        function increaseValue(id) {
            var element = document.getElementById(id);
            var currentValue = parseInt(element.value);
            if (currentValue < 300) { // Ensure the value does not exceed
300
                element.value = currentValue + 10; // Increase by 10 (or
adjust as needed)
            }
        }
    }
</script>

</body>
</html>

```

EXPLANATION OF HTML CODE :

This HTML document creates a user interface for a settings page, possibly for configuring a software application or a mission control system. The page includes options for general settings, mission default values, and area default values. Below is an explanation of the code structure and functionality:

1. Document Structure

- <!DOCTYPE html>: Defines the document as an HTML5 document.
- <html lang="en">: Specifies that the document language is English.
- <head>Section: Contains metadata about the document.
 - <meta charset="UTF-8">: Sets the character encoding to UTF-8.
 - <meta name="viewport" content="width=device-width, initial-scale=1.0">: Ensures the page is responsive on different devices.
 - <title>: Sets the title of the page.
 - <link rel="stylesheet" href="general setting.css">: Links an external CSS file (general setting.css) for styling.

2. Body Structure

- The body is divided into two main panels: the left panel for general and mission settings, and the right panel for units and area default values.

3. Left Panel (left-panel panel)

- General Settings:

- Theme Selection: A dropdown menu (<select>) allows the user to choose a theme (Light, Dark, Solarized).

- Language Selection: Another dropdown menu for selecting the interface language (English, Spanish, French, German).

- Sound Button: A button toggles the sound setting between "Sound: Off" and "Sound: On".

- Mission Default Values:

- Several settings for mission parameters, such as WP Speed, WP Altitude, Rails Speed, Rails Altitude, Track Family Interval, Circle Speed, Circle Altitude, and Timeout.

- Each setting has an input field (<input type="number">) for entering a value, and a button to increase the value by 10 units up to a maximum of 300.

4. Right Panel (right-panel panel)

- Units Settings:

- Dropdown menus for selecting units for distance (meters or feet), coordinates, and speed (m/s or knots).

- Area Default Values:

- Several area parameters like Exclusion, Containment, Min Depth, Max Depth, and Contacts.

- Each parameter has toggle buttons (styled as circles) that can be clicked to switch states, indicating whether a border or fill is applied.

5. JavaScript Functionality

- Theme Selection:

- Listens for changes in the theme dropdown and applies the selected theme to the body by adding/removing appropriate classes (dark-theme, solarized-theme).

- Sound Toggle Button:

- Toggles the text on the sound button between "Sound: Off" and "Sound: On" when clicked.

- Placeholder for additional logic to actually enable or disable sound.

- Value Increase Buttons:

- A function (increaseValue(id)) increases the numeric value of mission parameters by 10 units when the corresponding button is clicked, up to a maximum of 300.

- Toggle Buttons:

- A function (toggleButton(button)) toggles the active class on the circular buttons to visually indicate their state.

CSS CODE OF SETTING PANEL :

```
body {
  display: flex;
  margin: 0;
  font-family: Arial, Helvetica, sans-serif;
  height: 100vh;
  overflow: hidden;
  background-color: f4f4f4;
  color: 000;
  font-size: 13px;
}

.left-panel, .right-panel {
  background-color: f4f4f4;
}

/ Dark Theme /
body.dark-theme {
  background-color: 333;
  color: fff;
}

body.dark-theme .left-panel,
body.dark-theme .right-panel {
  background-color: 444;
}

/ Solarized Theme /
body.solarized-theme {
  background-color: 002b36;
  color: 839496;
}
```

```

body.solarized-theme .left-panel,
body.solarized-theme .right-panel {
    background-color: 073642;
}

/ Other existing styles /
.panel {
    padding: 20px;
    box-sizing: border-box;
    overflow-y: auto;
}

.left-panel {
    width: 40%;
    background-color: f4f4f4;
    padding-left: 5%;
}

.right-panel {
    width: 35%;
    background-color: f4f4f4;
    padding-left: 7%;
}

h1 {
    font-size: 20px;
    margin-top: 0;
}

.form-group {
    margin-bottom: 15px;
    width: 25%;
    font-family: Arial, Helvetica, sans-serif;
}

label {
    display: block;
    margin-bottom: 5px;
}

select, button, input {
    width: calc(100% - 20px);
    padding: 8px;
    box-sizing: border-box;
    margin-bottom: 10px;
}

button {
    background-color: 007bff;
    color: white;
    border: none;
    cursor: pointer;
    font-family: Arial, Helvetica, sans-serif;
}

```



```

}
button:hover {
    background-color: 0056b3;
}
.value-display {
    font-weight: bold;
}
.mission-section {
    margin-top: 20px;
}.mission-item, .unit-item, .area-item {
    display: flex;
    align-items: center;
    margin-bottom: 15px;
}
.mission-item label, .unit-item label, .area-item label {
    flex: 1;
    min-width: 150px;
}
.mission-item input, .unit-item input, .area-item input {
    flex: 2;
    margin-right: 10px;
}
.mission-item button, .unit-item button, .area-item button {
    flex: 1;
}
.circle-button {
    width: 20px;
    height: 20px;
    border-radius: 50%;
    border: 2px solid 000;
    background-color: fff;
    margin-right: 10px;
    cursor: pointer;
    text-align: center;
    font-family: 'Courier New', Courier, monospace;
}
.circle-button.active {
    background-color: orange;
}

```

EXPLANATION OF CSS CODE :

This CSS code is used to style a webpage with different themes and layout configurations, specifically designed for a settings page. The code is structured to provide different visual styles for a light theme, a dark theme, and a solarized theme. Below is a detailed explanation of the code:

1. General Layout and Styling

- body:
 - display: flex: Makes the body a flex container, which allows for flexible layout of its child elements.
 - margin: 0;; Removes any default margin.
 - font-family: Arial, Helvetica, sans-serif;; Sets the font for the entire page.
 - height: 100vh; : Sets the height of the body to fill the entire viewport height.
 - overflow: hidden; : Prevents any overflow content from being displayed outside the body.
 - background-color: f4f4f4; : Sets the default background color for the page to a light gray.
 - color: 000;; Sets the default text color to black.
 - font-size: 13px;; Sets the default font size.
- .left-panel, .right-panel:
 - background-color: f4f4f4; : Sets the background color of the left and right panels to light gray.

2.Theme-Specific Styles

- Dark Theme (.dark-theme):
 - Applies when the dark-theme class is added to the body.
 - background-color: 333; : Changes the body background color to a dark gray.
 - color: fff;; Changes the text color to white.
 - background-color: 444;; Changes the background color of the left and right panels to a slightly lighter gray.
- Solarized Theme (.solarized-theme):
 - Applies when the solarized-theme class is added to the body.
 - background-color: 002b36;; Changes the body background color to a dark blue-green (Solarized base color).
 - color: 839496;; Changes the text color to a lighter grayish-blue (Solarized text color).
 - background-color: 073642;; Changes the background color of the left and right panels to a slightly lighter version of the body background color.

3. Panel and Form Styling

- .panel:
 - padding: 20px; : Adds padding inside the panel.
 - box-sizing: border-box;: Ensures padding is included in the element's total width and height.
 - overflow-y: auto; : Allows vertical scrolling within the panel if content overflows.
- .left-panel:
 - width: 40%;: Sets the width of the left panel to 40% of the body.
 - padding-left: 5%;: Adds padding to the left side.
- .right-panel:
 - width: 35%; : Sets the width of the right panel to 35% of the body.
 - padding-left: 7%; : Adds padding to the left side.

4. Typography and Form Elements

- h1:
 - font-size: 20px;: Sets the font size for headings.
 - margin-top: 0;: Removes the top margin.
- .form-group:
 - margin-bottom: 15px;: Adds space below each form group.
 - width: 25%;: Sets the width of form groups to 25%.
 - font-family: Arial, Helvetica, sans-serif;: Ensures consistent font usage.
- label:
 - display: block;: Ensures labels are block elements.
 - margin-bottom: 5px;: Adds space below the label.
- select, button, input:
 - width: calc(100% - 20px);: Sets the width of these elements relative to their container, leaving 10px padding on each side.

- padding: 8px;; Adds padding inside the elements.
- box-sizing: border-box;; Includes padding and border in the element's total width and height.
- margin-bottom: 10px;; Adds space below each element.

- button:

- background-color: 007bff;; Sets the button's background color to blue.
 - color: white;; Sets the button text color to white.
 - border: none;; Removes the default border.
 - cursor: pointer;; Changes the cursor to a pointer on hover.

- button:hover:

- background-color: 0056b3;; Changes the background color to a darker blue when the button is hovered over.

- .value-display:

- font-weight: bold;; Makes the text bold.

5. Section and Item Styling

- .mission-section:

- margin-top: 20px;; Adds space above the mission section.

- .mission-item, .unit-item, .area-item:

- display: flex;; Arranges the elements inside each item as a flexbox.
 - align-items: center;; Vertically aligns items in the center.
 - margin-bottom: 15px;; Adds space below each item.

- .mission-item label, .unit-item label, .area-item label:

- flex: 1;; Sets the flex-grow property to 1, allowing the label to take up available space.
 - min-width: 150px;; Sets a minimum width for the labels.

- .mission-item input, .unit-item input, .area-item input:
 - flex: 2;; Allows the input fields to take up more space compared to the labels.
 - margin-right: 10px;; Adds space to the right of the input fields.

- .mission-item button, .unit-item button, .area-item button:
 - flex: 1;; Makes the buttons take up available space equally.

6. Circle Button Styling

- .circle-button:
 - width: 20px;; Sets the width of the button to 20px.
 - height: 20px;; Sets the height to 20px, making it a circle.
 - border-radius: 50%;; Makes the button circular.
 - border: 2px solid 000;; Adds a black border.
 - background-color: fff;; Sets the background color to white.
 - margin-right: 10px;; Adds space to the right of the button.
 - cursor: pointer;; Changes the cursor to a pointer on hover.
 - text-align: center;; Centers any text or content inside the button.
- .circle-button.active:
 - background-color: orange;; Changes the background color to orange when the button is active.