## **Multi-Step Form Extension Guide**

### **Introduction**

This guide explains how to add new steps to the multi-step form. It includes all necessary details, such as adding new form fields, setting up navigation, and handling the form’s progression. By following this guide, future developers will be able to easily extend the form by adding more steps as needed.

### **Structure of the Form**

The form consists of multiple steps, each inside a <div> with the class form-step. The form has a navigation system with buttons that allow users to go forward or backward between the steps.

#### **Current Steps Overview:**

1. **Step 1** - Search for a Service
2. **Step 2** - Choose a Service
3. **Step 3** - Set Your Budget
4. **Step 4** - Provide Your Location
5. **Step 5** - Select Additional Features
6. **Step 6** - Enter Your Details
7. **Step 7** - Form Completed

### **Steps to Add a New Step**

#### **1. Add the HTML for the New Step**

To add a new step, you need to create a new <div> element inside the <form> with the class form-step.

##### **Example:**

Let’s say you want to add a new step where users can enter their preferences for notifications.

<!-- Step 8: Notification Preferences -->

<div class="form-step" id="step8">

<h2>8- Notification Preferences</h2>

<div class="content-container">

<!-- Radio buttons for notification preferences -->

<div class="box-container">

<div class="box">

<input type="radio" id="emailNotification" name="notification" value="email" required>

<label for="emailNotification">Email Notifications</label>

</div>

<div class="box">

<input type="radio" id="smsNotification" name="notification" value="sms" required>

<label for="smsNotification">SMS Notifications</label>

</div>

</div>

<!-- Navigation buttons for step -->

<div class="button-container">

<button type="button" class="prev-btn" onclick="prevStep(7)">Back</button>

<button type="button" class="next-btn" onclick="nextStep(9)">Next</button>

</div>

</div>

</div>

#### **2. Add Navigation Logic**

Next, you need to add the logic for navigating between the steps. This is done using the nextStep() and prevStep() functions. Make sure to reference the new step in these functions.

##### **Example:**

In the nextStep() function, add a case for your new step:

function nextStep(step) {

if (validateStep(currentStep)) {

showStep(step);

if (step === 9) {

showModal();

}

}

}

#### **3. Update the showStep() Function**

The showStep() function controls the transition between steps. It needs to be updated to handle the new step.

function showStep(step) {

const currentStepElement = document.getElementById(`step${currentStep}`);

const nextStepElement = document.getElementById(`step${step}`);

currentStepElement.classList.add('blur');

currentStepElement.classList.add('slide-out');

nextStepElement.style.display = 'block';

nextStepElement.classList.remove('slide-in');

setTimeout(() => {

currentStepElement.classList.remove('active', 'blur', 'slide-out');

nextStepElement.classList.add('active', 'slide-in');

setTimeout(() => {

nextStepElement.classList.remove('slide-in');

}, 500);

}, 500);

currentStep = step;

}

#### **4. Validate the New Step**

If your new step includes form fields (such as radio buttons, checkboxes, etc.), you will need to validate them as well in the validateStep() function.

function validateStep(step) {

const currentStepElement = document.getElementById(`step${step}`);

const inputs = currentStepElement.querySelectorAll('input[required]');

for (let input of inputs) {

if (!input.value.trim()) {

alert('Please fill out all required fields.');

return false;

}

}

return true;

}

#### **5. Add CSS Styles for the New Step**

For consistency in design, make sure to add appropriate CSS styles for your new step in the style.css file.

css

Copy code

/\* Style for Step 8 \*/

#step8 {

display: none;

opacity: 0;

transition: opacity 0.5s ease-in-out;

}

#step8.active {

opacity: 1;

}

### **Complete Example Code for Adding a Step**

Let’s say you added a step to select notification preferences (Step 8). Here’s how it would look in the context of the full form.

#### **HTML:**

<!-- Step 8: Notification Preferences -->

<div class="form-step" id="step8">

<h2>8- Notification Preferences</h2>

<div class="content-container">

<div class="box-container">

<div class="box">

<input type="radio" id="emailNotification" name="notification" value="email" required>

<label for="emailNotification">Email Notifications</label>

</div>

<div class="box">

<input type="radio" id="smsNotification" name="notification" value="sms" required>

<label for="smsNotification">SMS Notifications</label>

</div>

</div>

<div class="button-container">

<button type="button" class="prev-btn" onclick="prevStep(7)">Back</button>

<button type="button" class="next-btn" onclick="nextStep(9)">Next</button>

</div>

</div>

</div>

#### **JavaScript:**

function nextStep(step) {

if (validateStep(currentStep)) {

showStep(step);

if (step === 9) {

showModal();

}

}

}

function prevStep(step) {

showStep(step);

}

function validateStep(step) {

const currentStepElement = document.getElementById(`step${step}`);

const inputs = currentStepElement.querySelectorAll('input[required]');

for (let input of inputs) {

if (!input.value.trim()) {

alert('Please fill out all required fields.');

return false;

}

}

return true;

}