<!DOCTYPE html>

<html>

<head>

<title>Phone Number Verification</title>

</head>

<body>

<h1>Phone Number Verification</h1>

<label for="phoneNumber">Phone Number:</label>

<input type="text" id="phoneNumber" />

<button onclick="sendOTP()">Send OTP</button>

<script>

function sendOTP() {

const phoneNumber = document.getElementById('phoneNumber').value;

if (phoneNumber) {

fetch('/api/verifyPhoneNumber', {

method: 'POST',

headers: {

'Content-Type': 'application/json'

},

body: JSON.stringify({ phoneNumber: phoneNumber })

})

.then(response => response.json())

.then(data => {

// Handle the response from the backend

console.log(data);

})

.catch(error => {

// Handle any errors

console.error(error);

});

}

}

</script>

</body>

</html>

1. Backend (C# using ASP.NET Web API):

First, make sure you have the necessary NuGet packages installed:

* Microsoft.AspNet.WebApi
* Microsoft.AspNet.Cors (to enable Cross-Origin Resource Sharing)

Next, create a new ASP.NET Web API project in Visual Studio and replace the contents of the Controllers/ValuesController.cs file with the following code:

I apologize for the incomplete response. Here's the complete code for generating the OTP and returning it from the backend:

using System;

using System.Net.Http;

using System.Net.Http.Headers;

using System.Threading.Tasks;

using System.Web.Http;

public class VerifyPhoneNumberController : ApiController

{

[HttpPost]

public async Task<IHttpActionResult> Post([FromBody] PhoneNumberRequest request)

{

string phoneNumber = request.PhoneNumber;

// Generate an OTP

string otp = GenerateOTP();

// You can use Gupshup's API to send the OTP to the user's phone number

// Here's an example using HttpClient (make sure to replace YOUR\_API\_KEY and YOUR\_SENDER\_ID)

using (HttpClient client = new HttpClient())

{

client.BaseAddress = new Uri("https://api.gupshup.io");

client.DefaultRequestHeaders.Accept.Clear();

client.DefaultRequestHeaders.Accept.Add(new MediaTypeWithQualityHeaderValue("application/json"));

// Replace YOUR\_API\_KEY and YOUR\_SENDER\_ID with your Gupshup credentials

string apiKey = "YOUR\_API\_KEY";

string senderId = "YOUR\_SENDER\_ID";

string message = $"Your OTP is: {otp}";

// Build the request body

var requestBody = new

{

method = "sendMessage",

send\_to = phoneNumber,

msg = message,

msg\_type = "text",

userid = apiKey,

auth\_scheme = "plain",

password = apiKey,

v = "1.1",

format = "json",

mask = senderId

};

HttpResponseMessage response = await client.PostAsJsonAsync("/v1/message", requestBody);

response.EnsureSuccessStatusCode();

return Ok(new { success = true, otp = otp });

}

}

private string GenerateOTP()

{

// Generate a random 6-digit OTP

Random random = new Random();

int otpNumber = random.Next(100000, 999999);

return otpNumber.ToString();

}

}

public class PhoneNumberRequest

{

public string PhoneNumber { get; set; }

}

Make sure to replace 'YOUR\_API\_KEY' and 'YOUR\_SENDER\_ID' in the code with your actual Gupshup API key and sender ID.

This code should allow your front-end to send a phone number to the back-end API endpoint /api/verifyPhoneNumber using a POST request. The back-end will then generate an OTP, send it to the provided phone number using Gupshup's API, and return the success status and OTP as a response. You can handle the response in the front-end and proceed accordingly.

Remember to configure the necessary routing and enable CORS (Cross-Origin Resource Sharing) if needed in your ASP.NET Web API project.

Let me know if you need any further assistance!

if gupshup provides functionality to generate otp then use that

Certainly! If Gupshup provides functionality to generate OTP, you can use their API for generating OTP instead of generating it locally. Here's the updated code that utilizes Gupshup's API for OTP generation:

using System.Net.Http;

using System.Net.Http.Headers;

using System.Threading.Tasks;

using System.Web.Http;

public class VerifyPhoneNumberController : ApiController

{

[HttpPost]

public async Task<IHttpActionResult> Post([FromBody] PhoneNumberRequest request)

{

string phoneNumber = request.PhoneNumber;

// Use Gupshup's API to generate OTP

using (HttpClient client = new HttpClient())

{

client.BaseAddress = new Uri("https://api.gupshup.io");

client.DefaultRequestHeaders.Accept.Clear();

client.DefaultRequestHeaders.Accept.Add(new MediaTypeWithQualityHeaderValue("application/json"));

// Replace YOUR\_API\_KEY with your Gupshup API key

string apiKey = "YOUR\_API\_KEY";

// Build the request body

var requestBody = new

{

method = "get",

userid = apiKey,

auth\_scheme = "plain",

password = apiKey,

v = "1.1",

format = "json",

mask = "true"

};

HttpResponseMessage response = await client.PostAsJsonAsync("/v1/otp/request", requestBody);

response.EnsureSuccessStatusCode();

var responseData = await response.Content.ReadAsAsync<GupshupOTPResponse>();

// Send the OTP to the user's phone number using Gupshup's API

string otp = responseData.OTP;

// Here you can call the Gupshup API to send the OTP to the user's phone number

// ...

return Ok(new { success = true, otp = otp });

}

}

}

public class GupshupOTPResponse

{

public string OTP { get; set; }

}

public class PhoneNumberRequest

{

public string PhoneNumber { get; set; }

}

In this code, the /api/verifyPhoneNumber endpoint will send a request to Gupshup's API to generate an OTP. The generated OTP is then returned as a response from the API. You can customize the code to include additional parameters or handle any error conditions that may arise during the API call.

Make sure to replace 'YOUR\_API\_KEY' in the code with your actual Gupshup API key.

Remember to configure the necessary routing and enable CORS (Cross-Origin Resource Sharing) if needed in your ASP.NET Web API project.

Let me know if you need any further assistance!