Automated Testimony Submission System

Setup & Configuration Guide

System Overview

This automated system allows users to submit testimony to the Colorado Legislature through a WordPress form. The system automatically fills out and submits the government portal on behalf of users, sends confirmation emails, and captures screenshots of submissions.

Part 1: AWS Infrastructure Setup

Step 1: Create S3 Bucket

1. Navigate to S3 in AWS Console
2. Click Create bucket
3. Bucket name: <bucket-name>
4. Region: us-west-2
5. Leave other settings as default and create
6. Open the bucket → Permissions tab → Bucket Policy

Add this policy to make screenshots publicly accessible:

```
{   "Version": "2012-10-17",  "Statement": [{    "Sid": "PublicReadGetObject",
"Effect": "Allow",    "Principal": "*",    "Action": "s3:GetObject",
"Resource": "arn:aws:s3:::<bucket-name>/*"  }]}
```

7. Go to Management tab → Lifecycle rules → Create rule
    - Rule name: delete-old-screenshots
    - Scope: prefix 'testimony/'
    - Action: Expire current versions after 90 days
8. Create folder: config/
9. Upload bill-positions.json to config/ folder

Step 2: Configure Amazon SES

10. Navigate to SES → Verified identities
11. Click Create identity
12. Choose Email address
13. Enter: info@<name>.org (or your notification email)
14. Check inbox and click verification link

***Note:** SES starts in sandbox mode. In production, request production access to send emails to any address.*

Step 3: Create Lambda Function

15. Navigate to Lambda → Functions → Create function
16. Function name: testimony-automation
17. Runtime: Node.js 20.x

18. Create function
19. Go to Configuration tab → General configuration → Edit
    - Memory: 1536 MB
    - Timeout: 2 minutes
20. Go to Configuration → Environment variables → Edit

Add these variables:

Variable	Value
SCREENSHOT_BUCKET	<bucket-name>
NOTIFICATION_EMAIL	info@<name>.org

21. Add Chromium Layer for browser automation
    - Go to Code tab → Layers → Add a layer
    - Choose Specify an ARN
    - ARN: arn:aws:lambda:us-west-2:764866452798:layer:chrome-aws-lambda:45
22. Configure IAM permissions (see next step)

Step 4: Configure IAM Permissions

23. Navigate to IAM → Roles
24. Find the role: testimony-automation-role-xxxxx (created with Lambda)
25. Click Add permissions → Create inline policy
26. Switch to JSON tab and paste:

```
{  "Version": "2012-10-17",  "Statement": [    {      "Effect": "Allow",
"Action": ["s3:PutObject", "s3:GetObject"],      "Resource":
"arn:aws:s3:::<bucket-name>/*"    },    {      "Effect": "Allow",
"Action": ["ses:SendEmail", "ses:SendRawEmail"],      "Resource": "*"    }  ]}
```

27. Policy name: testimony-automation-policy
28. Create policy

Step 5: Create API Gateway

29. Navigate to API Gateway → Create API
30. Choose REST API → Build
31. API name: testimony-api
32. Endpoint type: Regional
33. Create API
34. Actions → Create Resource
    - Resource name: submit
    - Enable CORS: Yes
35. Select /submit → Actions → Create Method → POST
    - Integration type: Lambda Function
    - Lambda function: testimony-automation
    - Use Lambda Proxy integration: Yes
36. Actions → Deploy API

  - Stage: prod

37. Copy the Invoke URL (needed for WordPress form)

Part 2: Lambda Code Deployment

Step 6: Prepare Lambda Deployment Package

38. Create a new directory locally: C:\Users\[username]\testimony-lambda
39. Open command prompt in this directory
40. Initialize npm project:

```
npm init -y
```

41. Install dependencies:

```
npm install playwright-core @sparticuz/chromium @aws-sdk/client-s3 @aws-sdk/client-ses
```

42. Create index.js file with the Lambda code (provided separately)
43. Create deployment package:

PowerShell:

```
del testimony-lambda.zippowershell Compress-Archive -Path * -DestinationPath
testimony-lambda.zip
```

Step 7: Upload to Lambda

44. Go to Lambda console → testimony-automation
45. Code tab → Upload from → .zip file
46. Select testimony-lambda.zip
47. Click Deploy

Important: *Always upload as zip. Editing code directly in Lambda console will remove node_modules.*

Part 3: WordPress/Elementor Form Setup

Step 8: Create Elementor Form

Create a new page and add an Elementor Form widget with these fields:

Field Type	Field ID	Settings
Checkbox	bills	Label: Select bill(s) to testify onOptions: HB26-1056|HB26-1056
Text	first_name	Required: Yes
Text	last_name	Required: Yes
Email	email	Required: Yes
Tel	phone	Required: Yes
Text	address_street	Required: Yes
Text	address_city	Required: Yes

Text	address_zip	Required: Yes
Select	testify	Options: In Person, Remotely via Zoom
Acceptance	authorized	Required: YesText: I authorize submission

Step 9: Configure Form Webhook

48. In Elementor form, go to Actions After Submit
49. Add Webhook action
50. Webhook URL: [Your API Gateway URL]/submit
51. Example: https://1n09pr6fl9.execute-api.us-west-2.amazonaws.com/prod/submit
52. Set Success Message: 'Thank you! Please check your email for confirmation or further instructions.'
53. Set Error Message: Same as success message (handles API Gateway timeout)

Part 4: Bill Position Configuration

Managing Bill Positions

Bill positions are stored in S3 and can be updated without code changes.

To Update Positions:

54. Navigate to S3 → <bucket-name> → config/
55. Download bill-positions.json
56. Edit in any text editor
57. Upload back to S3 (overwrites)
58. Changes take effect immediately

Example bill-positions.json:

```
{  "HB26-1056": "Against",  "HB26-1070": "For",  "HB26-1096": "Neutral",
"default": "For"}
```

Part 5: Monitoring & Troubleshooting

CloudWatch Logs

All Lambda execution logs are stored in CloudWatch:

- Navigate to CloudWatch → Log groups → /aws/lambda/testimony-automation
- View recent executions, errors, and debugging information
- Look for 'Bills to process', 'Position for [bill]', and 'Success' messages

Common Issues

Issue: API Gateway timeout after 29 seconds
Solution: This is expected. Lambda continues processing. User sees success message and gets email when complete.

Issue: No bills selected error
Solution: Check Elementor form field ID matches 'bills'. Verify checkbox options are formatted correctly.

Issue: Incorrect position submitted
Solution: Update bill-positions.json in S3. Ensure bill numbers match exactly.

Issue: Screenshots not accessible
Solution: Verify S3 bucket policy allows public read access. Check IAM permissions include s3:PutObject.

Maintenance & Updates

Updating Lambda Code

59. Make changes to index.js locally
60. Re-create zip package
61. Upload to Lambda
62. Test with a form submission
63. Check CloudWatch logs for errors

Adding New Bills to Form

64. Edit Elementor form checkbox options
65. Add new bill: HB26-XXXX|HB26-XXXX - Description
66. Update bill-positions.json in S3
67. Add entry: 'HB26-XXXX': 'For/Against/Neutral'
68. No code changes required