WaveScan Frontend Intern Assessment

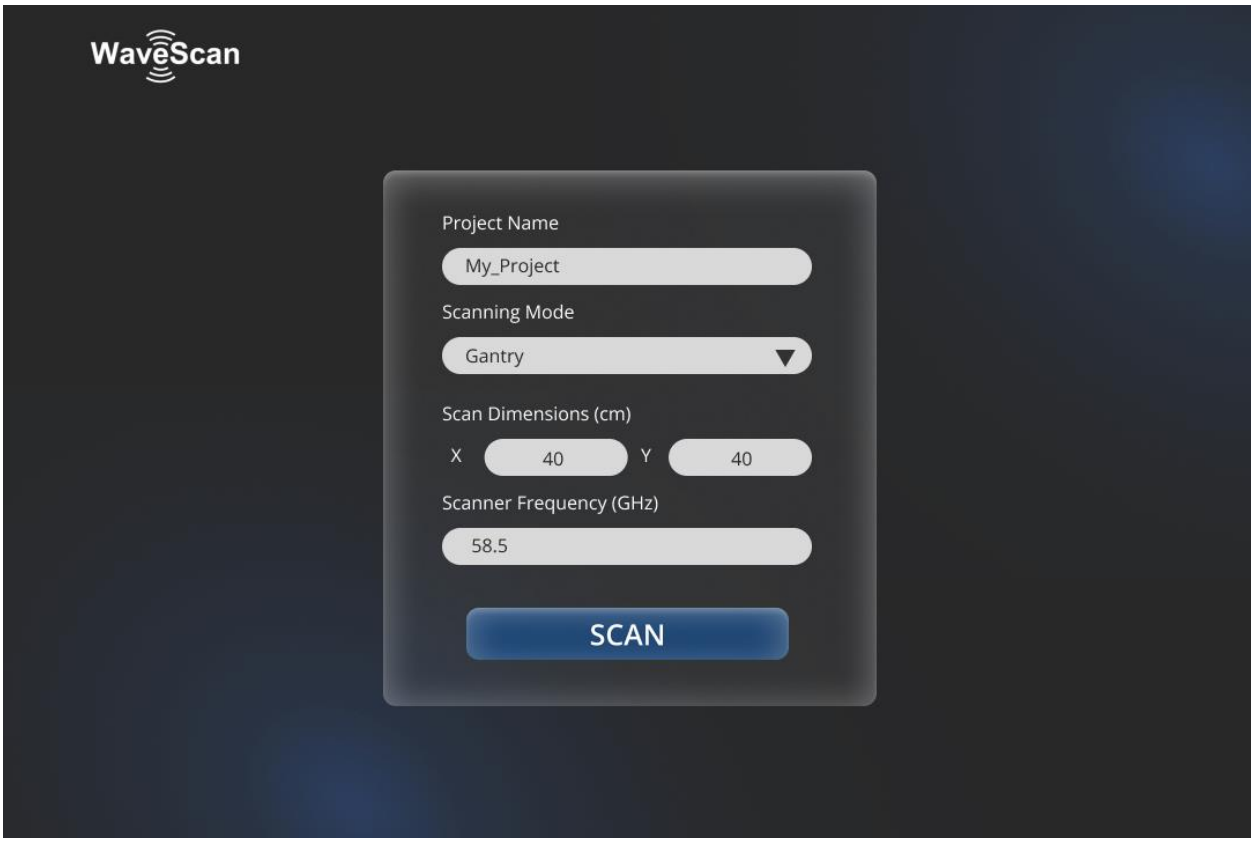
Task Outline: After numerous consultations with clients, you realize the need for a simple web application that takes in parameters for the scanner and posts it to a backend for validation. Thereafter, you will need to fetch and display the details of all the available scanners.

We primarily use React.js and Svelte at WaveScan. However, feel free to use any other framework of your choice (E.g., Angular, Vue).

Part A: Form Page to collect data

Build a page with a form that collects parameter details for the scanner and sends it to the server via a POST request.

You may refer to the design below for reference. However, feel free to improve upon it.



The form submission will return a 200: OK success response or a 400: Bad Request error response depending on the following data types and validation rules. The API is as follows:

API Endpoint: https://wavescan-internship.saurabhmudgal.repl.co/submitForm

Request Type: POST

Response: 200: OK if request is valid. Else, 400: Bad Request.

Request Body Parameter	**Expected Data Type**	**Validation Rules**
projectName	String	Has to be more than 3 characters
scanningMode	String	Accepted input: “GANTRY” “CRAWLER” “AUTO” “MANUAL” “ARM”
scanDimensionsX	Integer	Has to be >= 1
scanDimensionsY	Integer	Has to be >= 1
scannerFrequency	Float	Has to be >= 1, To 1 decimal place

Note: We do not expect you to implement form data validation, you can assume that it is handled by the server. The focus here is on displaying the correct UI based on the server response.

Part B: Handling Form Submission

Depending on the response from the server, display the following UI accordingly:

1. 200: OK Success → a) Success Page
2. 400: Bad Request Error → b) Error Page

a) Success Page

If the form data from Part A is valid, route the user to a success page displaying the data of all the available scanners. The list of all the scanners should be fetched via a GET request to the following API:

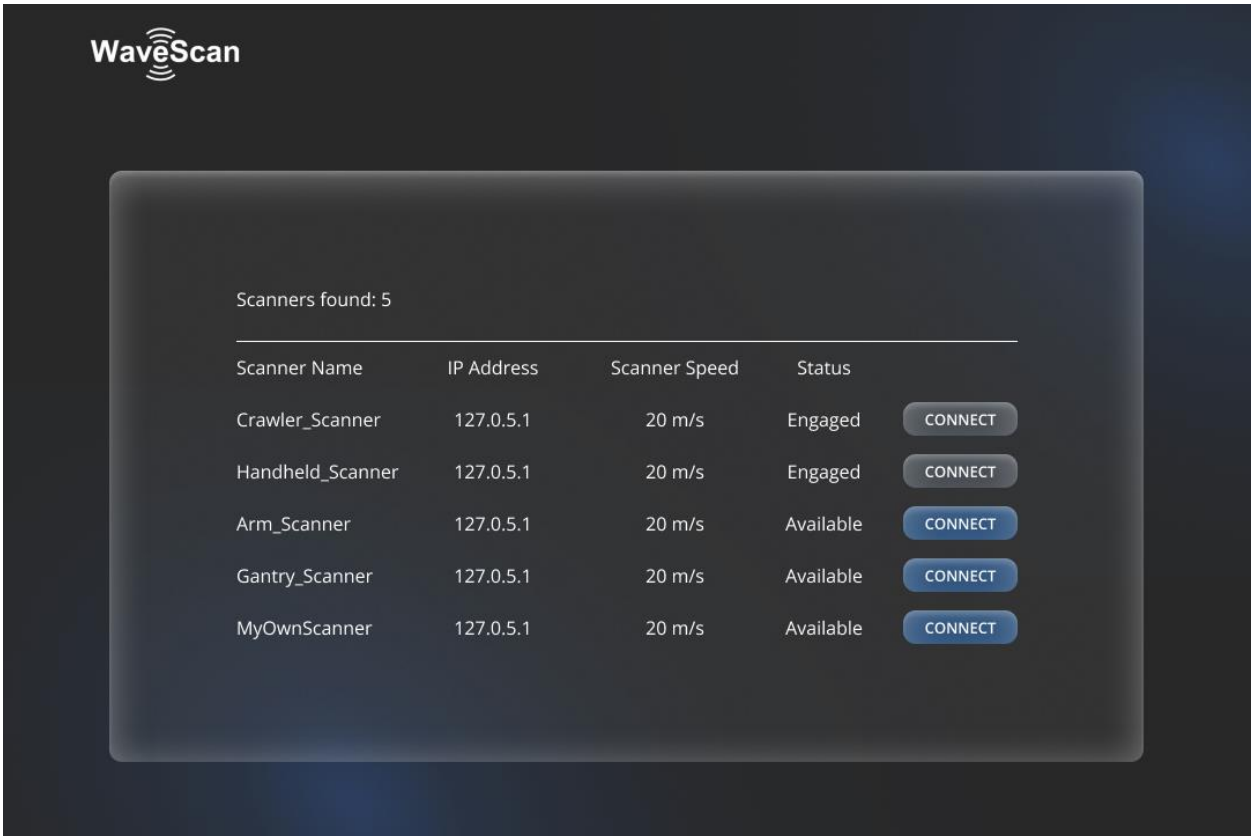
API Endpoint: https://wavescan-internship.saurabhmudgal.repl.co/success

Request Type: GET

Response: A list of ScannerDetails

ScannerDetails Object	
Data Field	**Value Type**
scannerName	String
ipAddress	String
scannerSpeed	Float
isAvailable	Boolean

You may refer to the design below for reference. However, feel free to improve upon it.



Note: The number of ScannerDetails objects in the response data for each request is randomized. You are also not required to implement the "Connect" button as shown above.

b) Error Page

Display a helpful error message to the user. Feel free to decide how you would inform the user that the form has not been submitted successfully.

Submission Instructions

Upload your code to GitHub and share the repository link with us. (Make sure that your repository is public!)

Additionally, we would love it if you could deploy your application and share the live site link with us. Don't know how to do that? Follow these simple instructions for deploying with GitHub Pages or Netlify.

If you have any doubts, feel free to reach out to our engineers at saurabh@wavescan.sg or darren@wavescan.sg. We look forward to seeing your work. All the best!