Interactive Form Validation

Phase 5 – Project Report

Contributors

Aruna M – 2023103079

Harsika V - 2023103588

Sharan Saminathan - 2023103609

Sri Bavan Akash S - 2023103627

Paril T – 2023103714

Abstract

The project titled “Interactive Form Validation” aims to design and develop an interactive and user-friendly web form that validates user input both on the frontend and backend. The system ensures proper data integrity by performing real-time validation using JavaScript and Bootstrap on the client side and Express.js on the server side.

Additionally, the backend handles API requests for username verification and form submissions, storing user data in a JSON file. The project demonstrates API communication, data handling, and deployment using Netlify and Render.

Tools and Technologies Used

The following tools and technologies were used for developing and deploying the project:

- HTML5, CSS3, Bootstrap – for building the responsive frontend user interface
- JavaScript – for implementing interactive validation logic
- Node.js and Express.js – for creating backend APIs
- Render – for hosting the backend server
- Netlify – for deploying the frontend web application
- GitHub – for version control and team collaboration

System Architecture

The architecture of the Interactive Form Validation system is based on a client-server model.

The frontend (hosted on Netlify) communicates with the backend API (hosted on Render).

When a user submits the form:

1. Validation occurs on the frontend (JavaScript).

2. The backend revalidates the input before storing.

3. Valid data is saved into users.json file.

4. A success message is displayed on the frontend.

Additionally, a dedicated API endpoint checks username availability to prevent duplicate accounts.

Module Description

The project is divided into the following key modules:

Form Validation Module:

- Handles client-side validation using JavaScript and Bootstrap to ensure all inputs are properly filled.

Username Availability Module:

- Interacts with the backend API to check if the entered username already exists before submission.

Form Submission Module:

- Submits validated data to the backend API endpoint securely.

Data Storage Module:

- Stores user details into a JSON file located in the backend server's /data folder.

Data Retrieval Module:

- Fetches and displays all stored users via the /API/users endpoint.

API Documentation

Endpoint	Method	Description	Example Response
/API/validate-username	POST	Checks if the entered username is available	{ "available": true }
/API/submit-form	POST	Submits the form data to the backend for storage	{ "success": true }

/API/users	GET	Fetches all stored user data from the backend	[{ "fullName": "...", "email": "..." }]

Challenges and Solutions

Challenge: Preventing automatic page refresh after form submission.

Solution: Used modal event handling to refresh only after the success message confirmation.

Challenge: Avoiding duplicate usernames.

Solution: Implemented an API-based username availability check.

Challenge: CORS errors during frontend-backend integration.

Solution: Enabled CORS middleware in the Express.js backend to allow secure cross-origin requests.

Challenge: Deploying frontend and backend on different platforms.

Solution: Configured Netlify for static frontend hosting and Render for backend Node.js deployment.

Future Enhancements

Replace JSON file storage with a cloud database such as MongoDB Atlas.

Implement user login and authentication features.

Add analytics and admin dashboard for monitoring user submissions.

Conclusion

The Interactive Form Validation project successfully demonstrates an end-to-end web application with real-time validation, API communication, and deployment.

The system ensures accurate data entry, an improved user experience, and secure backend integration.

Project Links

Frontend (Netlify): https://interactiveformwithvalidation.netlify.app/

Backend (Render): https://interactive-form-validation.onrender.com/api/users

GitHub Repository: https://github.com/SriBavanAkashS/Interactive-form-validation.git

Course: BE CSE – Anna University CEG