**Front-End UI/UX Mini Project**

**Project Title:** Form Validation Example

**Submitted by:**

* **Team members:** Thirisha Babu Velu

Aishwarya S

Tharika K

* **Roll Number:** 2460464

2460315

2460463

* **College E-mail id:** thirisha.babu@btech.christuniversity.in

aishwarya.s@btech.christuniversity.in

tharika.k@btech.christuniversity.in

**Course:** Front-End UI/UX Development

**Instructor Name:** Mr. Narendran

**Institution:** Christ University

**Date of Submission:**26.09.2025

**2.Abstract:**

* This project focuses on building a responsive and user-friendly form interface using HTML, CSS, JavaScript, and jQuery.
* The goal is to ensure accurate data collection through client-side validation techniques.
* The form includes various input types such as text, email, and password, and provides real-time feedback to users.
* The design emphasizes accessibility, clarity, and adaptability across devices using Bootstrap and custom styling.

**3.Objectives:**

* Create a structured form layout using semantic HTML.
* Implement validation rules for different input types using JavaScript and jQuery.
* Display contextual error messages to guide user input.
* Ensure the form is responsive and visually appealing across screen sizes.
* Enhance user experience with smooth transitions and accessible design.

**4.Scope of the Project:**

* Focused on front-end form design and client-side validation.
* No backend integration or server-side scripting.
* Responsive layout targeting desktop, tablet, and mobile devices.
* Emphasis on accessibility, usability, and clean error handling.
* Built using open-source tools and hand-coded HTML/CSS/JS.

**5.Tools & Technologies Used:**

|  |  |
| --- | --- |
| **Tool/Technology** | **Purpose** |
| HTML5 | Markup for form structure |
| CSS3 | Styling, layout, and responsive design |
| JavaScript | Client-side validation logic |
| jQuery | Simplified DOM manipulation and event handling |
| Bootstrap | Responsive grid and UI components |
| VS Code | Code editor |
| Chrome DevTools | Testing layout and debugging scripts |

**6.HTML Structure Overview:**

* Semantic tags used: <form>, <fieldsef>, <label>, <input>, <button>, <section>, <footer>.
* Structure:
* <header>: Title or logo
* <main>: Contains the form
* <form> with grouped fields using <fieldsef>
* <div> for error messages near each field
* <footer>: Copyright or contact info
* Acccessibility: All inputs linked to labels using for and id attributes.

**7.CSS Styling Strategy:**

* Internal CSS file is used.
* Layout:

1. Flexbox for vertical alignment of form fields.
2. Media queries for mobile responsiveness.

* Styling Techniques:

1. CSS Variables for theme control.
2. Smooth transitions for input focus and error states.
3. Error messages styled with red text and subtle animations.

* Mobile-first design: Prioritize small screens, then scale up.
* Optional: Bootstrap classes for grid layout and form controls.

**8.Key Features:**

|  |  |
| --- | --- |
| **Key Features** | **Description** |
| Custom Layout | Can arrange name, title and contact info in any way |
| Responsive design | Looks good on phones, tablets and computers |
| Semantic HTML | Uses clean tags that help with understanding and access |
| Print Styling | Prints neatly with special styles for paper |
| Hover effects | Links change when you move mouse over them |
| CSS theme variables | Easy to switch colours and styles using simple settings |

**9.Challenges Faced & Solutions:**

|  |  |
| --- | --- |
| **Challenges Faced** | **Solutions** |
| Wrong email or password | Used JavaScript to check patterns |
| Empty fields during submission | Added required field checks before submit |
| Making it work on all screen sizes | Used bootstrap and media queries |
| Styling errors without breaking layout | Applied CSS classes like .error and .valid |
| Giving instant feedback | Used oninput and keyup events |
| Keeping code clean and reusable | Wrote modular functions for each field type |

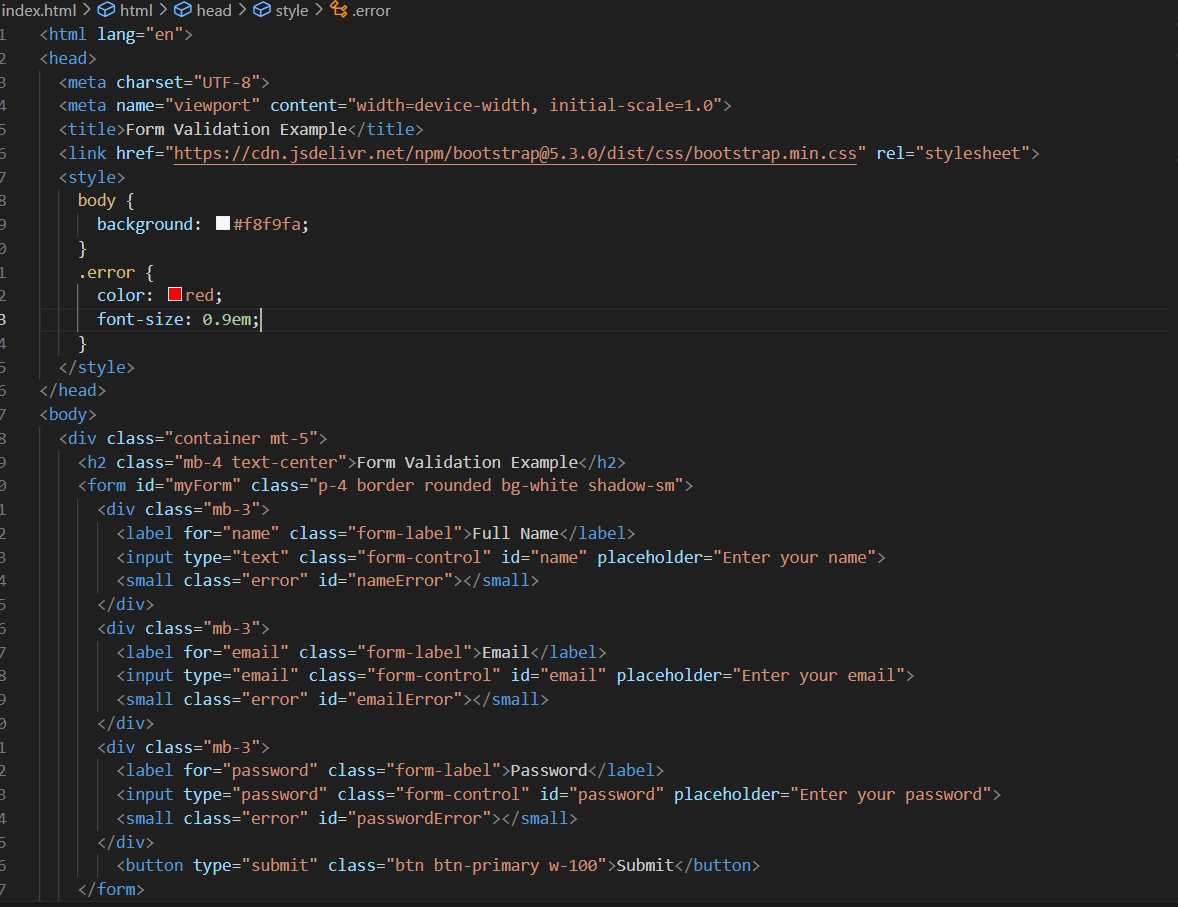
**10.Outcome:**

* A fully functional, responsive form that guides users to input valid data.
* Enhanced user experience through clear feedback and intuitive design.
* Clean, maintainable codebase with reusable components.
* Strong foundation for integrating backend validation or database storage.

**11.Future Enhancements:**

* Add password strength meter and show/hide toggle.
* Integrate CAPTCHA for spam protection.
* Include dark/light theme toggle.
* Add form analytics (e.g., field drop-off tracking).
* Backend integration for storing submissions.

**12.Sample Code:**

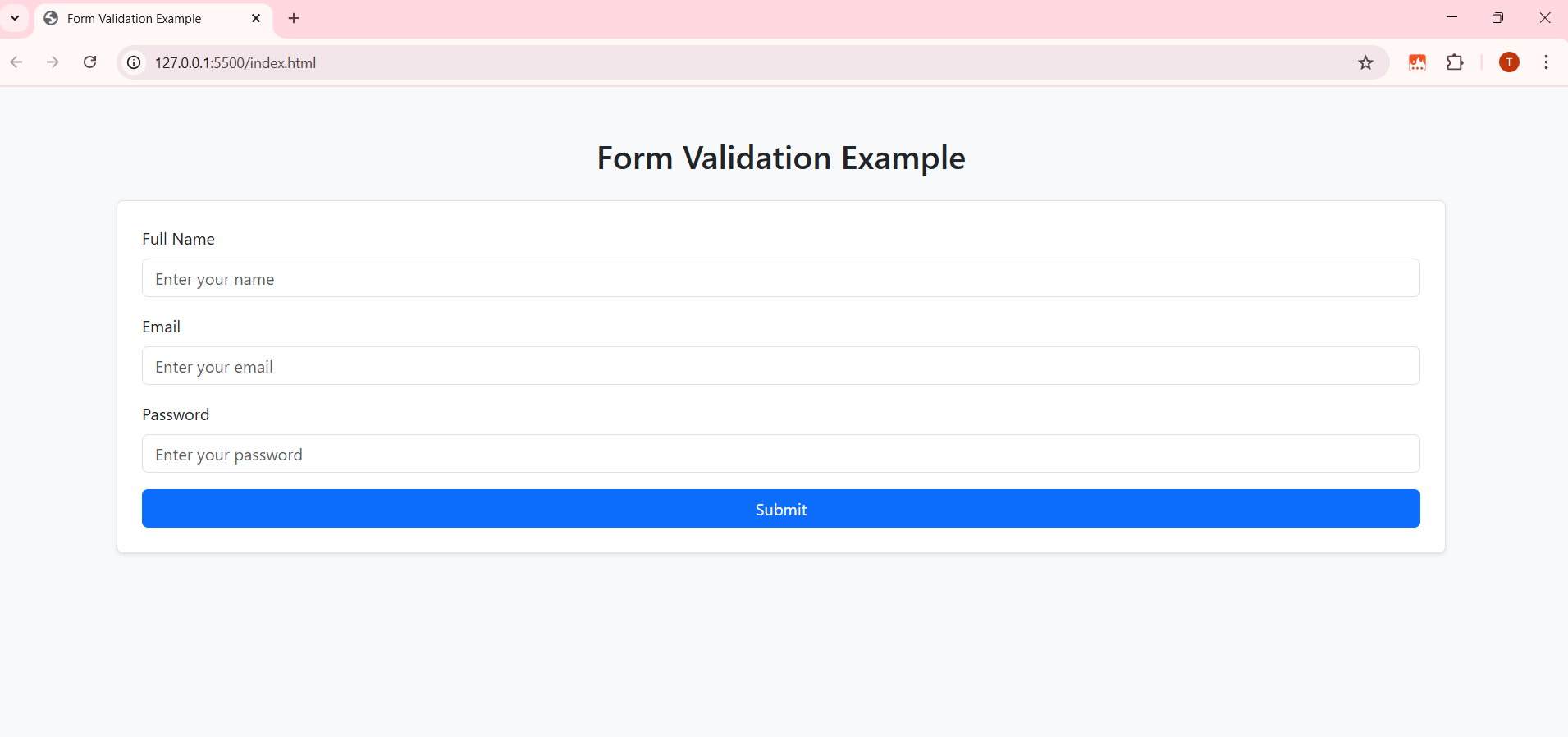


A screen shot of a computer program

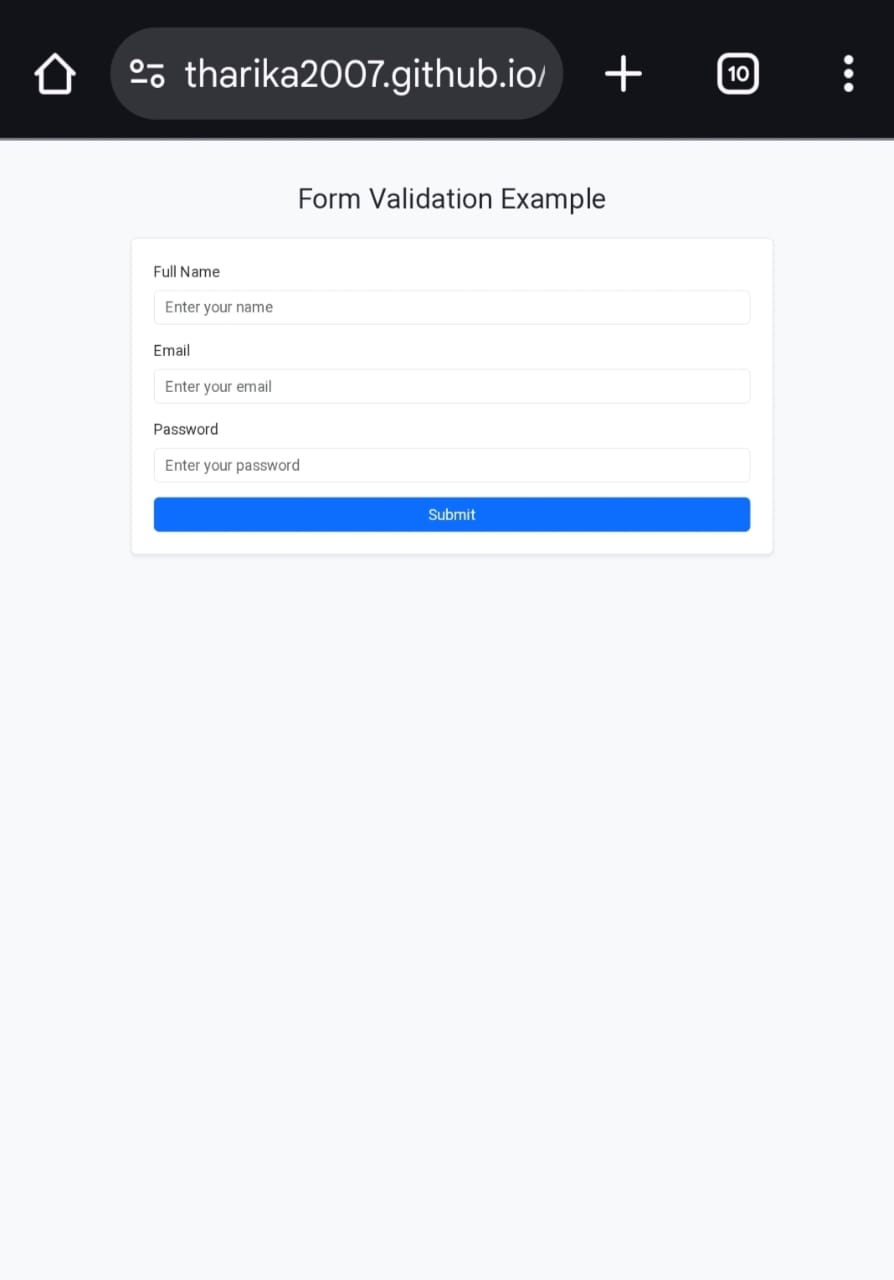
AI-generated content may be incorrect.

**13.Screenshots of Final Output:**

* Output as seen on laptop



* Output as seen on mobile phone



**14.Conclusion:**

* The Form Validation Example demonstrates the importance of user-friendly and error-resistant web forms in modern web development.
* By combining semantic HTML, responsive design, and client-side validation techniques, the project ensures accurate data collection and enhances user experience.
* This project lays a strong foundation for integrating backend functionality and expanding into more complex form workflows.

**15.References:**

* L&T LMS: https://learn.lntedutech.com/Landing/MyCourse