**Valid Bugs (bug id and screenshot) needed**

**🐞 Bug ID: BUG\_01 – File Upload Field Accepts Executable Files**

* **Title**: Unrestricted File Upload – Allows Arbitrary File Types Including Executables
* **Severity**: High
* **Priority**: High
* **Reported By**: *Kapil Rokaya*
* **Environment**:
  + **Browser**: Chrome 123.0.6312.106
  + **OS**: Windows 11 / Ubuntu 22.04
  + **URL**: <https://demoqa.com/automation-practice-form>

**Description**

The "Select Picture" file upload field lacks file-type restrictions and allows uploading of potentially harmful executable files like .exe, .bat, and .sh. There is no visible client-side or server-side validation enforcing MIME type restrictions, which could expose the application and users to security risks such as malware execution or social engineering attacks.

**Steps to Reproduce**

1. Visit: <https://demoqa.com/automation-practice-form>
2. Scroll to the “Select Picture” section
3. Click “Choose File”
4. Select any non-image file (e.g., test.exe, malware.bat, or script.sh)
5. Fill in other required form fields
6. Submit the form

**Expected Result**

* Only image files should be accepted (e.g., .jpg, .jpeg, .png)
* The system should reject other file types with a clear error message like:  
  *“Only image files are allowed.”*

**Actual Result**

* The form accepts and uploads non-image files, including .exe, .bat, and .sh, without any warning or error.
* There is no validation message or restriction preventing this.

**Impact / Risk Explanation**

* **Security Threat**: Attackers could upload harmful executables that, when accessed later (especially in a real app that stores/uploads), could infect systems or trick users into downloading malware.
* **Regulatory Compliance**: Lack of file-type validation may violate OWASP Secure Coding Practices and puts the app at risk of vulnerabilities like **Unrestricted File Upload (OWASP A10)**.
* **User Trust**: Users may unintentionally upload dangerous files if no validation is enforced.
* **Production Risk**: In a real production app with file storage or user profiles, this vulnerability could lead to **remote code execution (RCE)** or **information disclosure**.

A screenshot of a computer

AI-generated content may be incorrect.

**Suggested Fix**

* Implement client-side validation using accept="image/\*" on the <input type="file"> element
* Add server-side validation to enforce MIME type and file extension checks
* Reject files with dangerous extensions and provide a clear error message to users

**🐞 Bug ID: BUG\_02 – Date of Birth Field Accepts Future Dates**

* **Title**: Date of Birth Field Allows Selection of Future Dates
* **Severity**: Medium
* **Priority**: Medium
* **Reported By**: *Kapil Rokaya*
* **Environment**:
  + **Browser**: Chrome 123.0.6312.106
  + **OS**: Windows 11 / Ubuntu 22.04
  + **URL**: <https://demoqa.com/automation-practice-form>

**Description**

The Date of Birth field includes a calendar widget that allows users to select dates in the future, including years far ahead such as 2050. This contradicts the expected behavior, as a valid birth date should always be in the past. Allowing future dates results in **invalid user data** and creates the potential for **downstream logic errors** (e.g., age calculations, eligibility checks, etc.).

**Steps to Reproduce**

1. Open <https://demoqa.com/automation-practice-form>
2. Click on the **Date of Birth** input to open the calendar
3. Navigate to a **future year** (e.g., 2050)
4. Select any future date (e.g., 19 April 2050)
5. Fill out remaining required fields
6. Submit the form

**Expected Result**

* The calendar should prevent users from selecting any date **after the current day**
* The field should show a validation error like:  
  *“Date of Birth cannot be in the future.”*

**Actual Result**

* The calendar allows selection of dates in the future
* No validation or error message is displayed during or after form submission
* The form submits successfully with an illogical birth date

**Impact / Risk Explanation**

* **Data Integrity**: Invalid birth dates break logic dependent on age (e.g., age-based filters, legal compliance, school/college forms, etc.)
* **Business Logic Errors**: May lead to miscalculations or user profile issues if age is auto-calculated
* **Compliance Issues**: Forms handling sensitive data (e.g., educational institutions, government portals) are expected to validate personal data properly
* **User Experience**: Confuses users and allows entry of unrealistic or clearly incorrect data

**Screenshot**

A screenshot of a computer

AI-generated content may be incorrect.

**Suggested Fix**

* Add **client-side validation** using JavaScript to restrict future date selection
* Limit the date range in the calendar widget using attributes like maxDate = new Date()
* Add **server-side validation** to ensure no future dates are accepted even if client-side validation is bypassed

**🐞 Bug ID: BUG\_03 – Name Fields Accept Only Whitespace Characters**

* **Title:** First Name and Last Name fields accept only space characters
* **Severity:** High
* **Priority:** High
* **Reported By:** *Kapil Rokaya*
* **Environment:**
  + **Browser: Chrome 123.0.6312.106**
  + **OS: Windows 11 / Ubuntu 22.04**
  + **URL:** [**https://demoqa.com/automation-practice-form**](https://demoqa.com/automation-practice-form)

**Description**

The form accepts entries in the First Name and Last Name fields that contain only whitespace characters (e.g., spaces). This circumvents required field validation and allows the form to be submitted with invalid user input, leading to data integrity issues.

**Steps to Reproduce**

1. Navigate to <https://demoqa.com/automation-practice-form>
2. In the First Name and Last Name fields, press the space bar a few times (do not enter any actual characters)
3. Fill in all other mandatory fields with valid inputs
4. Submit the form

**Expected Result**

* The form should treat inputs containing only whitespace as empty
* A validation error should appear stating that the First Name and Last Name fields are required

**Actual Result**

* The form accepts whitespace-only inputs in the name fields
* No validation error is displayed
* The form submits successfully with invalid name data

**Impact / Risk Explanation**

* **Data Quality Issues:** User profiles created with only spaces in their name fields can break formatting and look unprofessional.
* **Backend Validation Errors:** Systems that consume this data may crash or throw exceptions if they attempt to process or parse blank names.
* **Business Logic Risks:** Downstream systems like user directories, certificates, email generation, or personalization may fail due to invalid name values.
* **Security Concerns:** Lack of input sanitization can be a broader indication of poor validation practices, possibly exploitable in other contexts

**Screenshot**

A screenshot of a computer

AI-generated content may be incorrect.

**Suggested Fix**

* Implement client-side input trimming and validation to detect and reject whitespace-only inputs
* Add server-side validation to ensure that submitted name fields contain non-whitespace characters
* Use input.trim().length === 0 logic to check for emptiness after trimming

🐞 **Bug ID: BUG\_04 – Special Characters Allowed in Name Fields**

* **Title**: First Name and Last Name Fields Allow Invalid Special Characters
* **Severity**: Medium
* **Priority**: High
* **Reported By**: *Kapil Rokaya*
* **Environment:**
* **Browser**: Chrome 123.0.6312.106
* **Operating System**: Windows 11 / Ubuntu 22.04
* **Device Type**: Desktop / Laptop
* **URL**: <https://demoqa.com/automation-practice-form>

**Description**  
The First Name and Last Name fields accept special characters such as @, #, !, and ~. This violates standard naming conventions and expected validation behavior. Allowing such input may result in invalid records, backend processing errors, or display issues across integrated systems.

**Steps to Reproduce**

* Navigate to: <https://demoqa.com/automation-practice-form>
* Enter K@pil# in the **First Name** field
* Enter Rok@#$!~ in the **Last Name** field
* Fill out all remaining mandatory fields
* Click the **Submit** button

**Expected Result**

* Form should restrict input to only valid alphabetic characters
* A validation error should be displayed like:  
  *“Please use only letters in name fields.”*
* Submission should be prevented until valid input is provided

**Actual Result**

* The form accepts invalid name values containing special characters
* The form submits successfully without displaying any validation message

**Impact / Risk Explanation**

* **Data Quality**: Invalid names may corrupt user databases or affect analytics
* **User Experience**: Accepting invalid data may confuse users or lead to errors later
* **Downstream Processing Errors**: Backend systems or reports may break or misbehave
* **Security**: Although not an injection vulnerability here, lack of sanitization increases risk surface

**Screenshot**

A screenshot of a computer

AI-generated content may be incorrect.

**Suggested Fix**

* Apply **client-side regex validation** to restrict invalid characters in name fields
* Suggested pattern: ^[A-Za-z\\s'-]+$ to allow only letters, spaces, hyphens, and apostrophes
* Add **server-side validation** to prevent bypass via direct request manipulation
* Display a clear error message to guide the user toward valid input

🐞 **Bug ID: BUG\_05 – Email Field Accepts Consecutive Dots in Domain**

* **Title**: Invalid Email Addresses with Consecutive Dots Are Accepted
* **Severity**: Medium
* **Priority**: Medium
* **Reported By**: *Kapil Rokaya*
* **Environment:**
  + **Browser**: Chrome 123.0.6312.106
  + **Operating System**: Windows 11 / Ubuntu 22.04
  + **Device Type**: Desktop / Laptop
  + **URL**: <https://demoqa.com/automation-practice-form>

**Description**  
The Email field accepts invalid addresses containing multiple consecutive dots (e.g., user@gmail...com). According to standard email format rules (RFC 5322), such patterns are invalid. Accepting them introduces the risk of data corruption, failed communication, or rejected emails by backend mail services.

**Steps to Reproduce**

* Navigate to: <https://demoqa.com/automation-practice-form>
* In the **Email** field, enter: user@gmail...com
* Fill out all other required fields correctly
* Click the **Submit** button

**Expected Result**

* The form should validate the email address format
* Email addresses with consecutive dots in the domain should be **rejected**
* A validation message like *“Invalid email address format”* should be displayed

**Actual Result**

* The form accepts and submits the invalid email without displaying any error
* No client-side or server-side validation is triggered

**Impact / Risk Explanation**

* **Data Quality**: Invalid emails can pollute databases and make records unusable
* **Communication Failures**: Backend systems may not deliver messages to invalid addresses
* **User Frustration**: Users may not receive important confirmations or updates
* **Compliance Issues**: Violates standard email validation rules (RFC 5322)

**Screenshot**

A screenshot of a computer

AI-generated content may be incorrect.

**Suggested Fix**

* Implement regex-based client-side email validation
  + Suggested regex: ^[a-zA-Z0-9.\_%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$ with additional logic to disallow .. sequences
* Add server-side validation to block malformed emails
* Display real-time error messages for better UX

🐞 **Bug ID: BUG\_06 – Email Field Accepts Invalid Local Part (Only Dots)**

* **Title**: Email Field Allows Invalid Email Structure with Only Dots Before the “@”
* **Severity**: High
* **Priority**: High
* **Reported By**: *Kapil Rokaya*
* **Environment:**
* **Browser**: Chrome 123.0.6312.106
* **Operating System**: Windows 11 / Ubuntu 22.04
* **Device Type**: Desktop / Laptop
* **URL**: <https://demoqa.com/automation-practice-form>

**Description**  
The email field permits inputs where the **local part** (portion before the “@”) consists solely of dots (e.g., .......@gmail.com). According to RFC 5322, this is not a valid email format. Accepting such entries leads to invalid data, failed email delivery, and potential issues in systems relying on email communication.

**Steps to Reproduce**

* Go to: <https://demoqa.com/automation-practice-form>
* In the **Email** field, enter: .......@gmail.com
* Fill all other required form fields
* Click the **Submit** button

**Expected Result**

* The email format should be validated
* An email with only dots in the local part should be **rejected**
* A clear error message should be displayed, such as: *“Invalid email format”*

**Actual Result**

* The form accepts and submits the invalid email address
* No validation error or message is shown

**Impact / Risk Explanation**

* **Email Delivery Failures**: Emails with this format will be undeliverable
* **System Integrity**: Invalid emails compromise data accuracy in user databases
* **Security Risk**: Input validation bypass may indicate broader flaws in form handling logic
* **UX Impact**: Users may not realize they’ve entered incorrect information and miss important communication

**Screenshot**

A screenshot of a computer

AI-generated content may be incorrect.

**Suggested Fix**

* Use stricter regex or validation logic to ensure the local part contains valid characters (alphanumeric and allowed symbols only)
* Prevent use of only dots or consecutive dots in the local part of the email
* Implement server-side validation in addition to client-side checks for robust email format enforcement