**Manual Checklist**

*1.UI*

* *Dropdown list things should be in a sequence manner*
* *Scrollbars, either vertical or horizontal, must not be apparent except when necessarily required.*
* *There should be an appropriate alignment of controls in the window.*
* *The size of buttons, menus, and text fields must be of proper estimation and should line up with different control features according to the page’s length and width.*
* *When some get access to the application notification of successfully popup/message should be generate in the page*
* *All the validation should work properly there should be validating message*
* *In every text field there should be some limitation if the limits exceed at that point, it must show a proper message under error information. (Email, phone number, password, name etc.)*
* *Session time out action is a significant check if any client exceeds the designated time span.*
* *Do all UI elements and content (text, images, animated GIFs, etc.) render on the page?*
* *Can the user navigate the UI?*
* *Are all links, menus, and submenus accessible, clickable and tappable? Are there any broken links?*
* *Is there a home link on every screen/page?*
* *Will the user be able to access all clickable elements (links, buttons, dropdowns, sliders, and boxes) accessible via keyboard only? Do all clickable/tappable elements receive focus via tabbing (i.e., pressing the tab key)?*
* *Do disabled fields and read-only elements receive focus via clicks, taps, and tabbing?*
* *Does the UI automatically place the cursor in the first (uppermost) text input field? Does this behavior match the specification?*
* *Are there any issues with the text content of buttons, fields, tooltips, messages, navigation items, and menus?*
* *Are there any spelling errors?*
* *Does the text content match the specifications and naming conventions?*

*2.For Buttons*

* *Should be enabled/disabled on page load*
* *What action should be carried out when button is clicked, if button submits form, then validation checks should be carried out on form fields.*

*3.calendar*

* Date format should be mm/dd/yyyy/<other format>
* In case of two date fields, “from” date should be behind “to” date
* Validation should check for leap year/month ending with 30/31 days

*3.Email*

* Valid Data: Alphabets, Numbers, -, \_, @, Dot
* Invalid Date: Special characters, Symbols, Leading and in-between white space
* Email address should have @ symbol
* At least one character should precede @ symbol
* @ Symbol should be followed by at least one character
* Followed by at least one 'dot'
* Dot should be followed by at least two characters
* Basic format should be [a@a.com](mailto:a@a.com)
* Application should not allow to use existing email address for registration

*4.Password*

* If text box is password field, then additional checks have to be carried out –
* Valid Data: Alphanumeric characters
* Invalid Data: To make password secure applications restrict using only characters or only number
* Should be masked when displayed in textbox
* Should be Encrypted while storing database
* Should not allow password text same as email address

*5.Zip code/Postal code*

* If text box is Zip code/Postal Code field then additional checks have to be carried out –
* MAX char limit restriction should be driven by the country selection
* Valid data should be driven by country selection. For example, only number for USA and number and alphabets for UK
* *If the text box is the country code field for the phone number, then additional checks have to be carried out –*
* *Max field length should be 3 digits*
* *Valid Data: whole numbers*
* *Invalid Data: Special characters, Symbols, Alphabets*

**Manual DB validation**

* *A column must not allow null unless a constraint states so.*
* *Whether data is saved in the right format and in the right way into the database.*
* *Check for password encryption.*
* *Verify how much does it take for a stored procedure to execute a command.*
* *What happens if wrong parameters are passed to stored procedure? Does application handle such use cases?*
* *Is data saved appropriately on form submit. Data should not be truncated when saved in database*
* *Verify time taken to execute a procedure. An improperly written procedure may result in longer response time*
* *All sensitive data i.e., password should be encrypted in data base*

**Manual security validation**

* *Make sure that all data is transmitted over the web using 'https' for instance payment information, passwords etc.*
* *Sensitive data should never appear in the URL.*
* *Once a user session is over, the same user should be prompted again to login.*
* *Keep a check on Denial of Service (Dos) attacks.*
* *Does logging in involve a two-step verification?*
* *After you’ve logged out, can you access your account (or any pages with sensitive data) without logging in?*
* *After you’ve changed the password, can you still log into your account with an old password?*
* *How many times can the user input an invalid password? Does the app lock the user out in case they have exceeded the number of attempts for entering the password?*
* *Error messages of any sort.*
* *Any pages that don’t require a login.*
* *In the source code of the page. If yes, is the* ***View Page Source / View Source Code*** *option disabled?*
* *Is there any sensitive data stored in cookies?*
* *Is sensitive data of any kind still accessible if some functionality of the app is not working?*
* *Are there traceable log files for storing important information? Does the app update these files as specified?*
* *Does the app use HTTPS / SSL?*
* *Does the app use encryption when handling sensitive data (including user credentials, user bio, credit card information, etc.)?*
* *If there are session values in the address bar, does the app encrypt these values?*
* *Does the app encrypt cookie data?*
* *What happens if you refresh the page or click/tap Back when a transferring sensitive data (e.g., credentials, payment info, etc.) is mid-progress?*
* *Can the user use your app after the session has expired?*
* *Can a user access the functionality that is only available to other roles (e.g., can a regular user access admin-only features)?*
* *How does the app handle an SQL injection?*
* *All sensitive data (password, credit/debit card, security answer) should transport over https protocol*
* *Once password is changed, then use should not be allowed to login using old password*
* *No sensitive data should be displayed in URL of application*
* *URL modification should not allow access privilege to under privileged users. For example, a user should not be able to access admin pages by just launching the admin page URL*
* *One user is logged out or session is expired then user should not be able to gain access to application using browser back. Application should force user to login*
* *No sensitive data should be logged in application log files*
* *Are multiple failed operations restricted by captcha to control denial of service (DOS) attack?*