**1. A Single UI page with Multi Browser Compatible and Mobile-friendly Responsive Layout**

The client requires you to design and develop a browser-based user and ticket registration service compatible with popular desktop browsers (e.g. Chrome, IE, Firefox & Safari) and mobile device browsers (e.g. IOS, Android and Windows). In addition the system must be capable of adapting and re-rendering its layout to help provide an optimal user experience regardless of the viewing device size or orientation.

**2. Database Design and Validation**

All data entered by the user MUST be verified and validated prior to being posted to the server and confirmation email sent to the user. Validation must include the prevention of invalid or unexpected characters being entered (review KEY events) and/or being copy and pasted (review BLUR or CHANGE events). As part of this requirement your system must be able to accept, validate and appropriately store (e.g. data type, size, index, etc.) the following information:

* \*Email Address (unique – the **DATABASE** must include constraints that prevent email duplication. Your system must also include **CLIENT SIDE** email address format validation (e.g. AA@AA.AA; you must also include rules that prevent users from entering any numeric values as part of the TLD etc.),
* \*Confirm Email Address (**CLIENT SIDE** validation: same as the email address entered in the Email Address field above)
* \*Title (**CLIENT SIDE** validation: use a drop-down menu to ensure validity),
* First Name (**CLIENT SIDE** validation: A-Z, a-z including: `SPACE` and `DASH`, min/max chars: 2/25),
* \*Last Name (see First Name),
* Landline No (**CLIENT SIDE** validation: 0-9 including: `SPACE`, `DASH`, ‘ROUND BRACKETS` and `PLUS`, min/max chars: 6/12),
* Mobile No (see Landline No),
* \*Address Line1 (**CLIENT SIDE** validation: A-Z, a-z, 0-9 including: `SPACE`, `SLASH` and `DASH`, min/max chars: 6/40),
* Address Line2 (see Address Line1),
* \*Town (**CLIENT SIDE** validation: A-Z, a-z including: `SPACE` and `DASH`, min/max chars: 4/40),
* \*Postcode (**CLIENT SIDE** validation: A-Z, 0-9 including: `SPACE`, min/max chars: 7/9),
* \*Country (**CLIENT SIDE** validation: A-Z, a-z including: `SPACE` and `DASH`, min/max chars: 4/40 Or a drop down menu to ensure validity and accuracy),
* \*Adult Tickets (**CLIENT SIDE** validation: drop down menu)
* Child Tickets (see Adult Tickets, however children under 13 MUST be accompanied by at least 1 adult)
* \*Privacy Policy (**CLIENT SIDE** Validation: check box) a) confirming registrant is 16 or over, registrant aged 13 to 15 must have guardian consent. b) read, understood and accepted the terms and conditions.
* Registration Date and Time (a **SERVER SIDE** generated date/time index)
* Password (a **SERVER** generated string that applies rules for password complexity)

*Please note: Fields marked ‘\*’ are considered mandatory. However, despite ‘Landline’ and ‘Mobile’ not being marked mandatory it is important that either one or both fields contain a valid contact number for the form to be considered valid (this is in addition to any other validation rules).*

**3. HCI and UX Design**

Ideally the system should allow the user to quickly and easily enter the required data (regardless of the device) and receive immediate, dynamic and helpful details indicating success or that a validation error has occurred (e.g. by highlighting the specific input and providing clear feedback on how to resolve the problem). In addition, the system must provide immediate details on the state of the email being entered (e.g. if the address would cause a duplicate record) supported by an appropriate message indicating the issue and how it can be resolved.

Once the form has been completed and fully verified on the CLIENT SIDE the system should automatically enable a SUBMIT button allowing the user to complete the registration by POSTING the data to the server and confirmation page send to the user along with the details registered, registration data and time, and a unique complex password for editing the registration details later.

Once the data has been sent - the server will carry out reasonable **SERVER SIDE** validation (including re-formatting all text fields to proper case and trimming whitespace, etc.).

**If no errors occurred the system will:**

* Create a complex password
* Store the re-formatted data including the created password and record an accurate registration date and time
* Compose a confirmation message consisting of all re-formatted data

**If an error does occur:**

* Compile a reasonable response regarding the problems and provide suggestions on how the issues could be corrected

Confirmation of **SERVER SIDE** success (or failure) must be sent back to the user with the additional information (e.g. a formatted version of the submitted data, the **SERVER SIDE** generated password and the registration timestamp).

To improve the UX and avoid information having to be re-entered the user’s registration page should not lose state (meaning all data entered by the user must remain, or at least be re-populated automatically by the system). If the submission was successful, the system must prevent the user re-submitting the same data twice.

Your page design(s) should be stable and any updates (including feedback, etc.) should be well integrated.

**4. Branding**

Maintain consistency (wherever possible) with the imagery and branding used on the existing Glastonbury Festivals website. Colour, Font and layout can be improved to provide a good user experience.

**5. Development**

You are expected to fully comment each line of code (denoting its purpose, how it works and what the outcome is). Optimise your code so that it becomes smaller size, consumes less memory and improves loading time and rendering speed. In addition, you are free to use or reuse jQuery plugins, snippets of code and published services providing you make clear reference to them as long as you are being able to fully explain their workings as well as identifying any possible issues. Although the user interface must be limited to 1 page you are free to add pages to support any **SERVER SIDE** services you feel are appropriate.