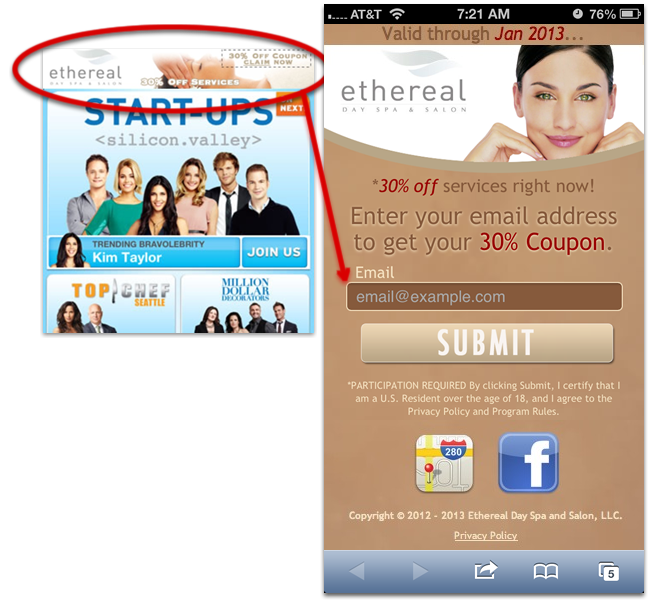
Please answer these questions in-line, in the whitespace just below each where applicable. Code file root folder should be zipped together and attached to your reply email.

Environment/technology requirements: valid HTML5, unobtrusive jQuery (without jQuery UI or any other libs), CSS3, PHP (no perl, no RoR or other libs), MySQL. Interfaces and dev tools are up to you. Must work well iPhone, Android, Win8 Phone, and be testable in Safari on Mac or Chrome on PC.

Please write all code samples as elegantly as possible. Legibility is also key; I have to be able to understand your code without reading the comments.

All JavaScript, HTML5 and CSS3 must actually work correctly in my browser. I will test. PHP and MySQL samples \*WILL NOT\* be tested for syntactic perfection and bug-free execution, but all middleware and backend concepts must meet logic muster. Pseudo-code is acceptable for PHP and MySQL components, as long as you observe proper syntactic structures and reserved words as applicable. Images have been provided in the zip.



Consider this lead-capture form. Users will land here after clicking a banner image on a partner website or from within a partner mobile app.

1. More data often correlates to more value. Aside from Email address, what are some other obvious data points that we can automatically collect?
2. Write out the HTML5 code for this page. Note that the file is index.php so you have server-side components if you need them. Include your additional data points from question #1 above. (Nothing to provide inline here.)
3. Write out all the necessary JS as an include-file – name it script.js or something semantically sound. Validate the email address using jQuery and regular expressions. Bind the submit handler of the form to the validation routine in an unobtrusive way; properly cancel the form submit when JS validation fails. Use an alert box to give the user feedback about any errors. Paste the snippet from jQuery selector through regex test of JS validation here.
4. Write out all the necessary CSS3 as an include-file – give it a semantically sound filename. (Nothing inline here.)
5. Write a SQL statement that creates the DB table that you would need in which to store these leads. Show your consideration for data retrieval optimization, for example: I don't like \*just\* storing a date/time timestamp, I like also storing a date-only column and an hour-only (0-23) column for quicker grouping. Write all SQL into a file named something semantically similar to "install.sql" and include in zip.
6. Many concepts can be addressed by both client-side and server-side code. Explain which concepts are better supported by PHP rather than JavaScript, and why.  
     
     
     
   Which concepts are better off in JavaScript over PHP?  
     
     
     
   Are there any data points we may want to collect that cannot be addressed in both JS and PHP? If so, what are they?  
     
     
     
   Write all these important code concepts into your index.php & script.js files.