

# Front-end developer Test - Front-end developer- Basic Test #1 - Instructions.pdf

**The Goal of this test is to assess your skills as a front-end developer, please use the landing\_page.html that is included and apply the following tasks. If you complete the first part, continue to the second part which is more advanced.**

Area	Task	Score	Total Allowed	Comments
<b>Page Setup</b>				
	Setup HTML5 Boilerplate		5	
	Add jQuery Library		5	
	Add External Stylesheet		5	
	Add External JS to contain all JavaScript		5	
<b>Design</b>				
	Add Bootstrap to make it responsive		25	
	Style Form: Remove list bullets, align fields into 2 columns, style input width according to their expected lengths.		5	
	Test that design is consistent in IE9+, Chrome, Firefox, Safari, iOS, Android		5	
<b>JS - On Page Load</b>				
	Hide company field and setup event on commercial radio input to show company field when this radio button is selected.		5	
	Phone Input - Only allow numbers to be entered.		5	
	Zip Code Input - Only allow numbers and maximum of 5 digits.		5	
<b>JS - On Form Submission</b>				
	Setup regular expression on email address to validate that its a real email address, throw error if not match.		5	
	Compare the values in Email Address and Confirm Email Address and make sure they match on submit. If not, display an error message. Disable paste function on confirm email address field.		5	
	If State value is equal to NY, CA, or MA - show a message on submit saying that we are unable to ship wine to those states and hide submit button. If different state is chosen, onblur - check against eligible states and if can ship to then re-enable submit button.		5	
	Setup regular expression on phone number to validate that its a real US phone number, throw error if not match.		5	
	Setup HTML5 require attribute on the following fields: email, first name, last name, address 1, city, state, zip, phone. Use modernizr for feature detection and offer backup JavaScript for browsers that dont support HTML5 input attributes. Highlight fields that are missing values.		5	
	Get all values from form inputs and store in an array. Hide completed form and use values in array and format as (below) on the page in place of the form.		10	
	Rewrite State Select and remove states we cannot ship to (MD, PA)		10	
	Add logging to check values/statements in console.		10	
	Set a cookie after successful form submission. After page refresh, load page - read cookie and hide form. Display message that the form has already been filled out and submitted.		10	
	Use HTML5 geolocation to set State value when the page has loaded.		10	
			10	
<b>Bonus Totals</b>		<b>0</b>	<b>60</b>	

## Front-end developer Test - Front-end developer - Advanced Test #2 - Builds Upon Basic

Area	Task	Score	Total Allowed	Comments / Example
<b>Page Setup</b>				
	Write Javascript using Object Literal Design Pattern			
	Setup GruntJS for JS and CSS tasks	0	25	
	Install necessary plugins to do below tasks utilizing package.json and NPM	0	25	
	Setup CSS as SASS	0	10	
	Add task for jshint to check JavaScript for errors before compiling	0	25	
	Add task to minify JS file for production	0	10	
	Add task to watch files and build anytime there is a change and reload page	0	15	
	Add task to build files for production to different directory	0	10	
	Add task to build SASS files into CSS files	0	10	
	Pull bootstrap module into SASS and build out 1 CSS file	0	10	
	Add task to build minify CSS files for production	0	10	
	<b>If using React for API interaction, setup webpack and babel to compile code</b>	0	25	
	-- Rewrite using ES6 API's where possible	0	25	
	<b>Totals</b>	<b>0</b>	<b>200</b>	
<b>JS - Forms and APIs</b>				
	Utilizing the Direct Wines Zip Code lookup API, on change of Zip code input, do an ajax call to the API (passing in zip code) and pull back values for city and state. Then assign those values to the city form and state form inputs	0	100	<a href="http://www.wsjwine.com/api/reference/jsp/rest/address_lookup.jsp">http://www.wsjwine.com/api/reference/jsp/rest/address_lookup.jsp</a>
	Utilizing the Direct Wines Email lookup API, on change of email address input, do an ajax call to the API (passing in email address) and pull back and display in the form using handlebarsjs	0	100	<a href="http://www.wsjwine.com/api/reference/jsp/rest/profile.jsp">http://www.wsjwine.com/api/reference/jsp/rest/profile.jsp</a>
	Utilizing the Direct Wines offer lookup API which returns details about a product that we are selling, use a front-end framework such as React, handlebars, or Angular and wire up a view using the data on top of the form.	0	100	<a href="https://www.wsjwine.com/api/offer/9183007">https://www.wsjwine.com/api/offer/9183007</a>
	Show only products from the above example that include this property: colourName: "Red"	0	20	
	<b>Setup above challenges using ReactJS</b>	0	100	
	<b>Totals</b>	<b>0</b>	<b>420</b>	
<b>JS - Working with a Cloud DB API</b>				
	Keep in tact all of the previous validation code	0	40	
	Setup Firebase or MongoDB Cloud database to store information	0	50	
	Add logging to all form values	0	10	
	Change submit form button to loading bar after it has been pressed and while waiting for response from cloud DB. Then switch to - thank you for submission.	0	50	
	Add values from form to cloud database after each submission	0	50	
	Setup handlebars and pull JSON from Firebase or MongoDB Cloud DB for all form submissions and display on page	0	50	
	<b>Totals</b>	<b>0</b>	<b>250</b>	
	<b>Test Totals</b>	<b>0</b>	<b>870</b>	
	Score out of 100%		0.00%	