The purpose of this project is to demonstrate basic programming ability in a simple web project. It will test basic HTML Markup, CSS, MVC, AJAX, Javascript, JQuery, and SQL using the Entity Framework. Much of the project has been stubbed out, you’re just being asked to fill in the missing pieces.

You are encouraged to use Google as a research tool, if needed, but you may not ask a friend for help over phone/chat/IM/etc. If you need to ask for help, please ask the interviewers.

When finished, you should have a basic website which allows viewing and manipulating DB record of contact information. You will be asked to demonstrate the completed project to the interviewers & discuss your code with them.

Please write the code as if you were actually writing it on the job (following what you consider to be best practice). The focus of this practice should be quality, not speed. There is no time limit, however, the overall project should be simple enough to complete within an hour or so. If time or other constraints impede you from following your own best practices, simply make a note of how you would have preferred to have accomplished the task & explain it to the interviewers upon completion of the project.

Screenshots of an example completed project are included at the end of this document. It is not required that your end project look exactly the same, but the functionality should be similar.

Please complete the following functionality in the Interview.sln project.

Unless a specific technology is specified (i.e. AJAX, Javascript, etc), you’re welcome to complete each task in whatever way you feel most comfortable (client-side, server-side, etc).

1. Create the action methods need to render the links on \_Layout.csthml
2. Create an Entity Framework data model to connect to the Interview.mdf database in the App\_Data folder.
3. Make the ContactGrid page display the data for all contacts in the database.
4. Implement creating a new contact and editing an existing one using the same ContactForm page. It should require the first and last name. Any update of the form should be done asynchronously. Upon successful creation, the grid should be shown including the new contact.
5. Implement deleting a contact. The delete must be done asynchronously. Upon successful deletion, the grid should be shown without the deleted contact. A success/failure message must be displayed on the screen.
6. Implement viewing a contact. There should be no submit button, and all data displayed must be read-only.