**DynamoDB Schema**

**Tables:**

1.Customer

id(email);

first\_name;

last\_name:

phone\_number;

address\_ref;  
2.Address

id;

apt;

building;

residental\_city;

residental\_state;

street;

zip\_code;

deleted;

**Operations:**

**1.Get on customer:** Return customer information based on email(id);

**2.Post on customer:** First make a post on address table, from this call we can get the address reference. Then insert this address reference with original data to customer table.

**3.Put on customer:** If update information doesn’t contain address information, just update related field. If contains address update, then first check if there exists the corresponding address in address table, if so, update customer’s address\_ref field. If not, create a new address entry and update customer’s address\_ref.

**4.Delete customer:** Delete the corresponding customer entry;

**5.Post, Get and Put on address:** Normal create, get and update on address. The UUID is generated by smartystreets.

**6.Delete on address:** Set the deleted field to true; This process has now been deprecated as the use case of deleting an address is now handled either by update address and changing the reference

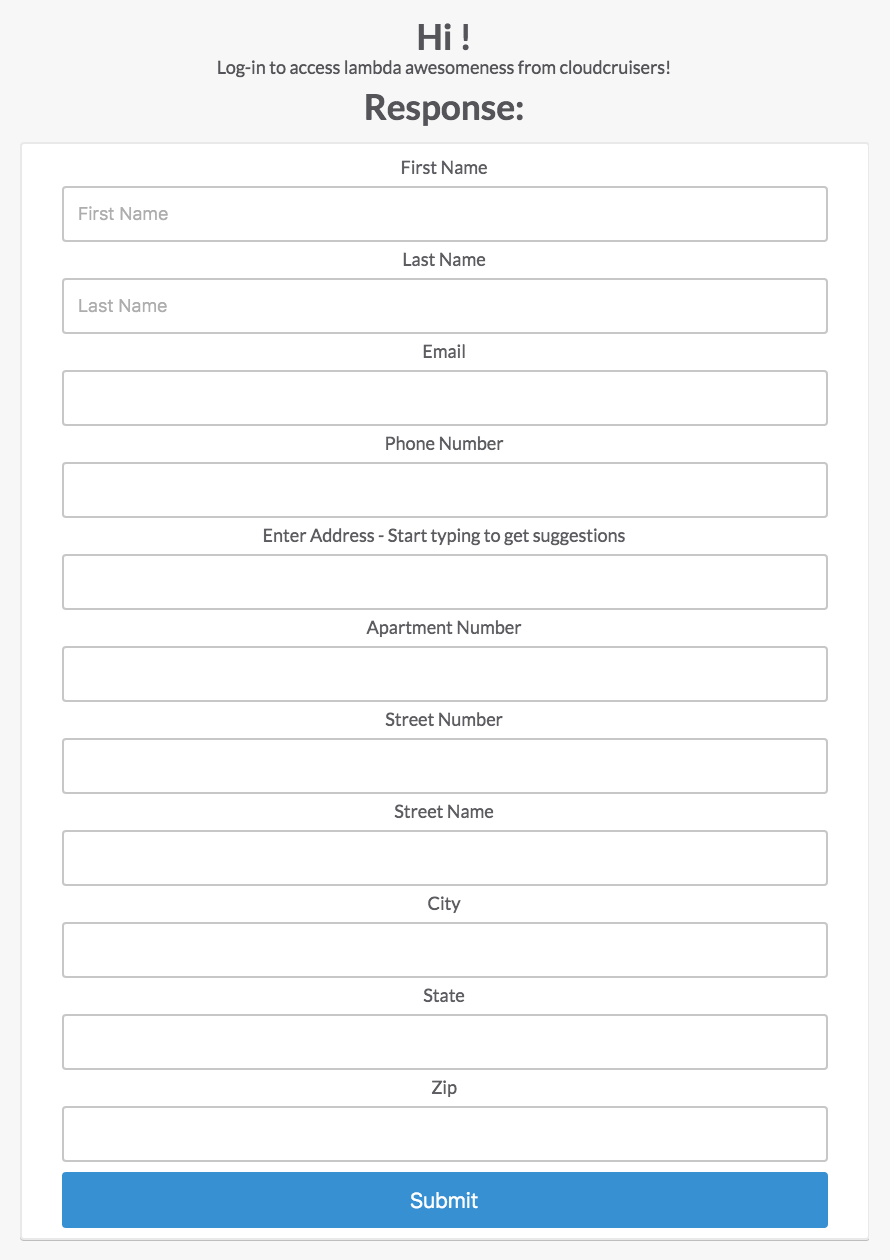
**UI**

1.index.html: This page is the customer registration page. It requires all the fields that has been described in dynamodb tables. A basic email, lastname, firstname regex validation is included. Also, address auto completion, validation and verification is implemented using smartystreets API.

2.controller: After user entered all the required fields, and clicked submit button. The controller will make a POST call on API gateway through this URL and then create a new customer: <https://j3z4xwt1ch.execute-api.us-west-2.amazonaws.com/prod/customers>.

UI Smarty streets call :  
The html page calls the smart streets API by a plugin. As the user begins typing the apartment no, street name in the address box, the autocomplete presents a list of suggestions from where the user can select. This is done by a JS function.

Once the address is selected, another function extracts the apartment number and the street name and populates the required fields. Same goes for city , state and Zip code. Once populated , the submit button issues a call to the API gateway to issue the POST command.

****