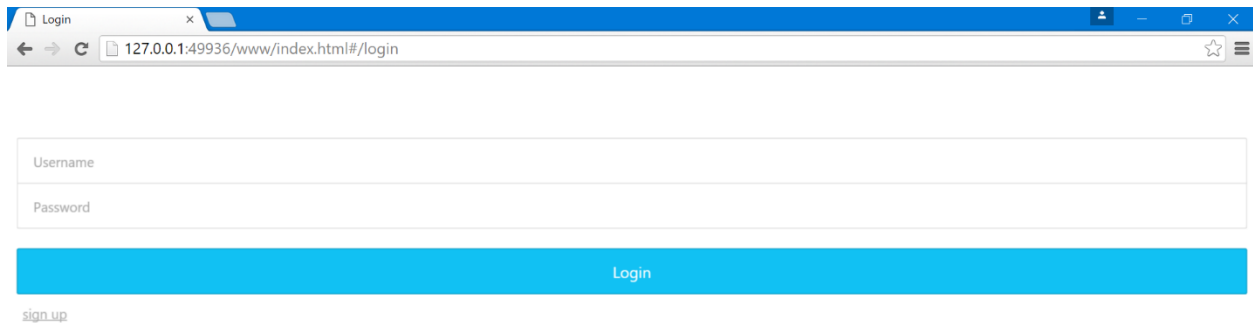


# Code Samples

Login screen: **click on singn up**



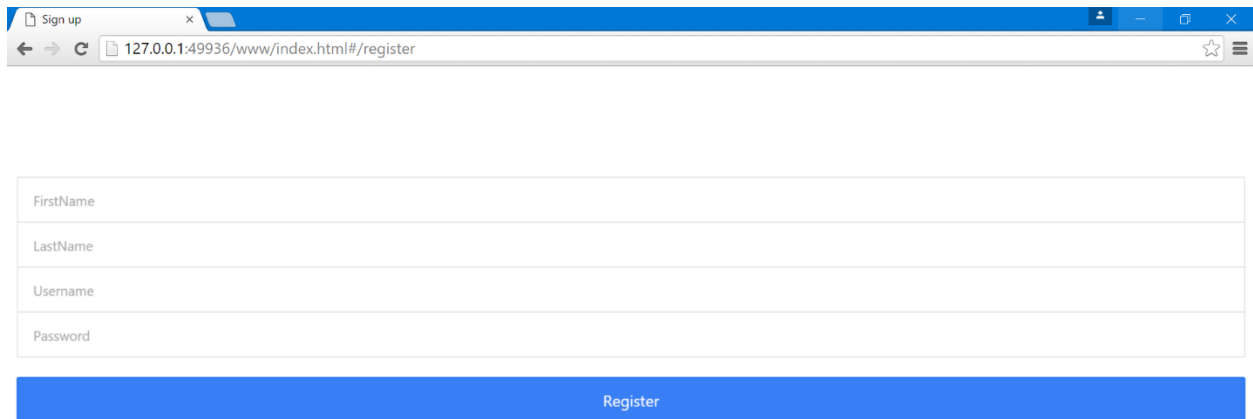
Username

Password

Login

[sign up](#)

Enter registration details



FirstName

LastName

Username

Password

Register

# Code Samples

Click on register

Sign up

127.0.0.1:49936/www/index.html#/register

vijay kumar

tummala

vij23049

...

Register

Search the web and Windows

12:23 PM 3/11/2016

You will be redirected to login page

Login

127.0.0.1:49936/www/index.html#/login

vij23049

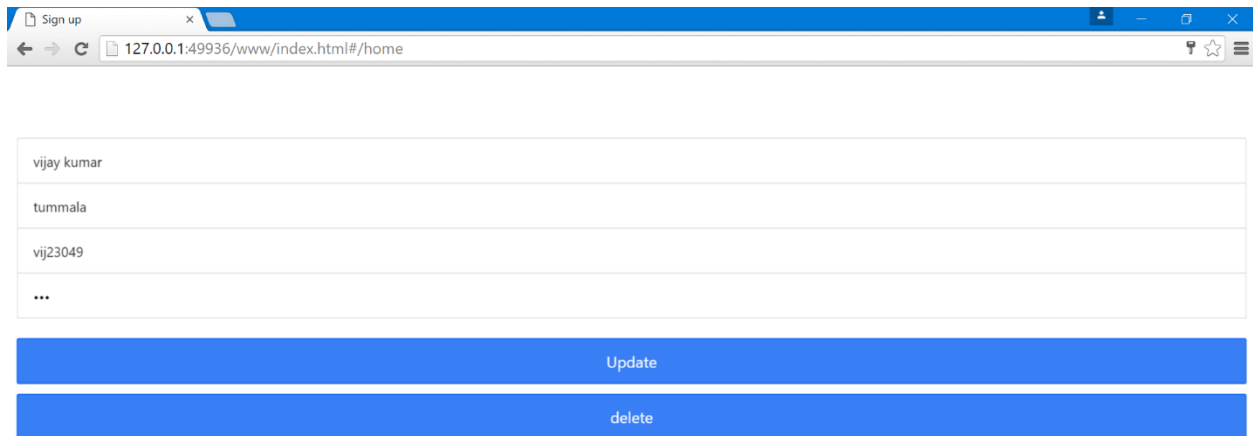
...

Login

[sign up](#)

# Code Samples

On success full login you will be redirected to home page with your details



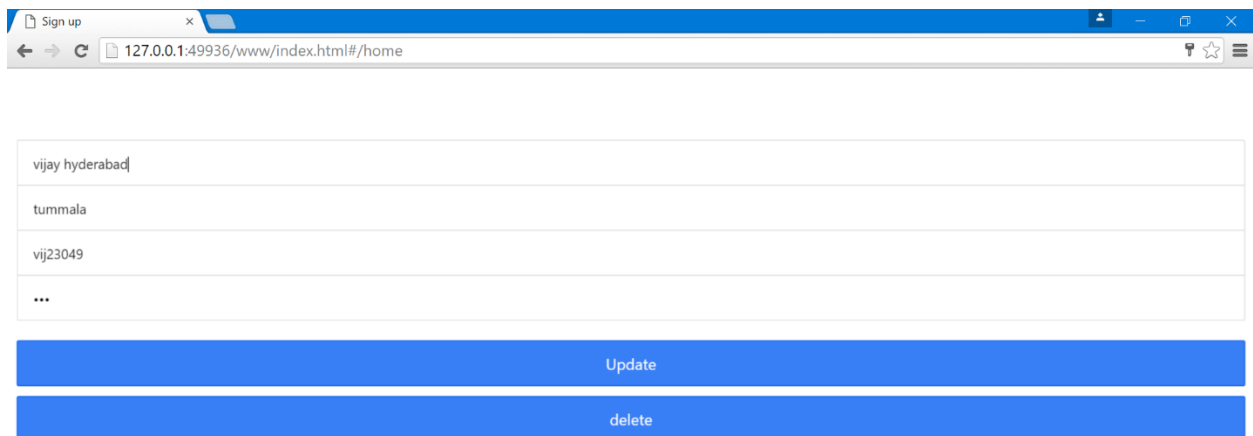
A screenshot of a web browser window. The address bar shows the URL `127.0.0.1:49936/www/index.html#/home`. The page displays a form with four input fields: "vijay kumar", "tummala", "vij23049", and "...". Below the form are two blue buttons: "Update" and "delete".

vijay kumar
tummala
vij23049
...

Update

delete

edit data and press update



A screenshot of a web browser window, similar to the one above. The address bar shows the URL `127.0.0.1:49936/www/index.html#/home`. The page displays a form with four input fields: "vijay hyderabad", "tummala", "vij23049", and "...". Below the form are two blue buttons: "Update" and "delete".

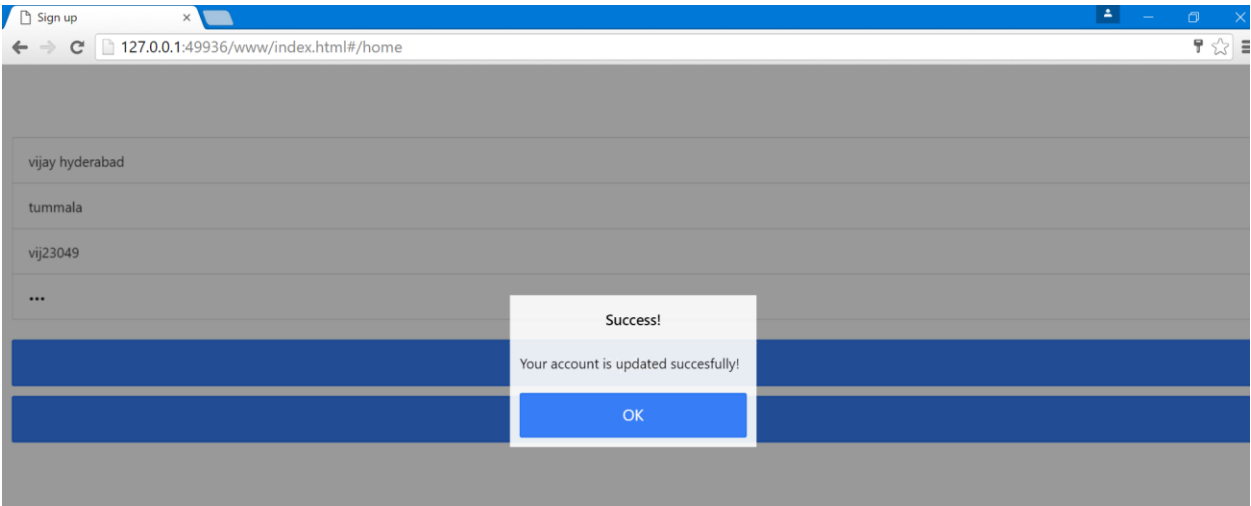
vijay hyderabad
tummala
vij23049
...

Update

delete

# Code Samples

Data is updated success fully



# Code Samples

The same is reflected in DB

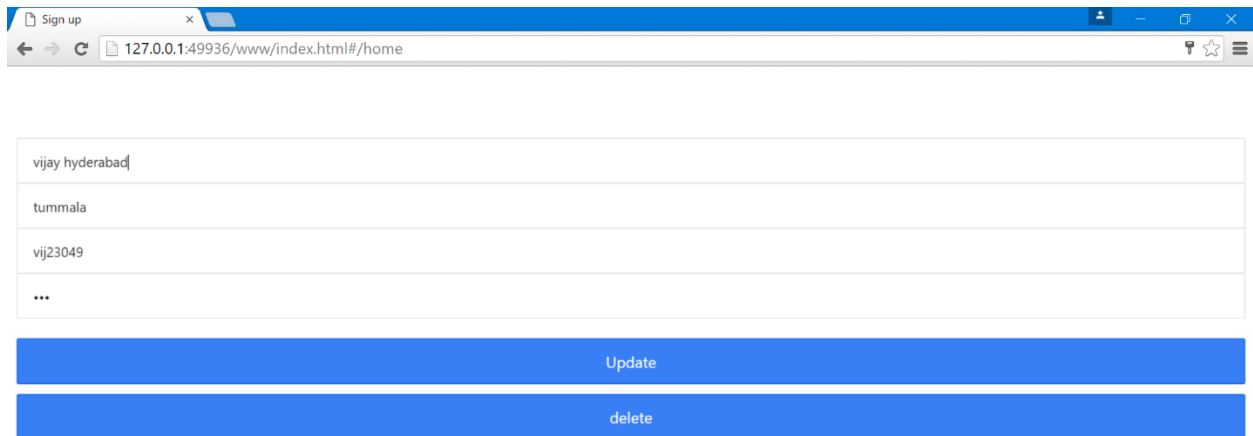
The screenshot shows a web browser window with the URL <https://mlab.com/databases/lab7/collections/userdata>. The page displays the MongoDB Atlas interface for the 'userdata' collection. The 'Documents' tab is selected, showing a single document in list view. The document contains the following fields:

```
{
  "_id": {
    "$oid": "56e30d87aebde02258a55639"
  },
  "firstname": "vijay hyderabad",
  "password": "123",
  "lastname": "tummala"
}
```

The document is displayed in a code editor with a scrollbar on the right. The browser's address bar shows the URL, and the page title is 'Collection: userdata'. The interface includes navigation tabs for 'Documents', 'Indexes', 'Stats', and 'Tools'. A search bar at the top allows for filtering documents. The document is shown in a light blue box with a close button (X) and a copy button (C) on the right.

# Code Samples

Re-login to DB and press DELETE

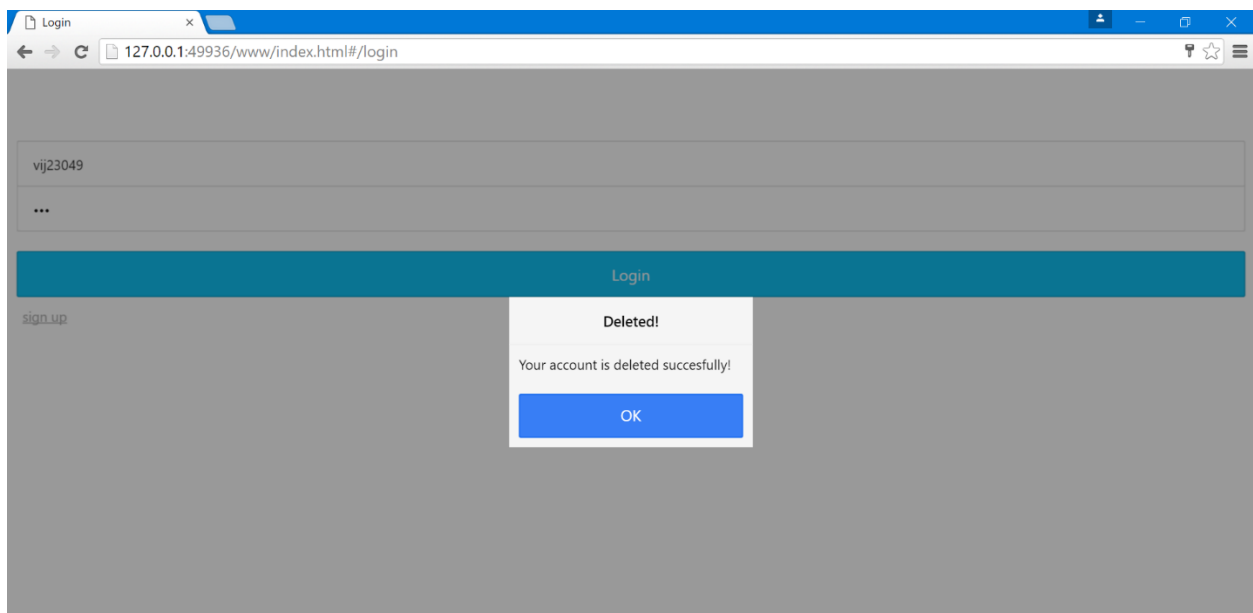


vijay hyderabad
tummala
vij23049
...

Update

delete

Data is deleted successfully



Deleted!

Your account is deleted succesfully!

OK

sign up

Login

vij23049
...

# Code Samples

Record is deleted successfully.

The screenshot displays the mLab MongoDB web interface. The browser's address bar shows the URL `https://mlab.com/databases/lab7/collections/userdata`. The page header includes the mLab logo and navigation links for 'WELCOME' and 'PLANS + FEATURES'. The main content area is titled 'Collection: userdata' and features a navigation bar with tabs for 'Documents', 'Indexes', 'Stats', and 'Tools'. Below the navigation bar, there are two buttons: 'Delete all documents in collection' (with a red 'X' icon) and 'Add document' (with a green '+' icon). A search bar is located below these buttons, containing the text '--- Start new search ---'. The main content area is titled 'All Documents' and displays the message '[None at this time]'. The browser's tab bar at the top shows several open tabs, including 'GitHub - Vijay-ASE/AS', 'CS551 - 2016 Spring', 'android game molecu', 'chain reaction game a', and 'GitHub'.

# Code Samples

```
lab8_ase/pom... web.xml loginMongo.java LoginService... Apache Tomca... CORSFilter.java testservice... Assg7/pom.xml »
1 package org.ase;
2
3 import com.mongodb.BasicDBObject;
10
11
12 public class loginMongo {
13
14     public String loginmongo(String userName) {
15
16         MongoClientURI uri = new MongoClientURI("mongodb://root:password@ds037824.mongolab.com:37824/lab7");
17         MongoClient client = new MongoClient(uri);
18         DB db = client.getDB(uri.getDatabase());
19         DBCollection users = db.getCollection("userdata");
20         BasicDBObject whereQuery = new BasicDBObject();
21         whereQuery.put("username", userName);
22         DBCursor docs = users.find(whereQuery);
23         System.out.println("here");
24         while(docs.hasNext()){
25             DBOBJECT doc = docs.next();
26             String b = doc.get("username").toString();
27             if(b.equals(userName)){
28                 System.out.println("1");
29                 client.close();
30                 return doc.toString();
31             }
32         }
33         client.close();
34         return null;
35     }
36
37 }
38
```

```
lab8_ase/pom... web.xml loginMongo.java LoginService... Apache Tomca... CORSFilter.java
1 package org.ase;
2
3 import javax.ws.rs.GET;
9
10 @Path("/login")
11 public class LoginService {
12
13     @GET
14     @Path("/{username}")
15     public Response getMsg(@PathParam("username") String username) {
16
17         loginMongo login = new loginMongo();
18         String output = login.loginmongo(username);
19         if(null!= output){
20             return Response.status(200).entity(output).build();
21         }else{
22             return Response.status(400).entity("fail").build();
23         }
24
25     }
26
27 }
```



# Code Samples

```
angular.module('starter.services', [])
  .service('LoginService', function($q, $http) {
    return {
      loginUser: function(name, pw) {
        var deferred = $q.defer();
        var promise = deferred.promise;

        $http({
          method: 'GET',
          url: 'http://localhost:8080/lab8_ase/rest/login/'+name,
          contentType:"application/json"
        }).success(function(data){
          if (name == data.username && pw == data.password) {
            localStorage.setItem("firstname", data.firstname);
            localStorage.setItem("lastname", data.lastname);
            localStorage.setItem("username", data.username);
            localStorage.setItem("password", data.password);
            localStorage.setItem("id", data._id.$oid);

            deferred.resolve('Welcome ' + data.username + '!');
          } else {
            deferred.reject('Wrong credentials. ');
          }
        })
        promise.success = function(fn) {
          promise.then(fn);
          return promise;
        }
        promise.error = function(fn) {
          promise.then(null, fn);
          return promise;
        }
        return promise;
      }
    }
  })
})
```