

PRACTICAL NO. 4

4.1 Installing and using Contacts plugin

4.2 Installing and using Device plugin

4.3 Installing and using Accelerometer plugin

4.1 Installing and using Contacts plugin

This plugin is used for accessing the contacts database of the device. In this tutorial we will show you how to create, query and delete contacts.

Step 1 - Installing Contacts Plugin

To install this plugin, we need to open the **command prompt** window and run the following code.

```
cordova plugin add cordova-plugin-contacts
```

Step 2 - Add Buttons

The button will be used for calling the **createContact** function. We will place it in the **div class = "app"** in **index.html** file.

```
<button id = "createContact">ADD CONTACT</button>  
<button id = "findContact">FIND CONTACT</button>  
<button id = "deleteContact">DELETE CONTACT</button>
```

Step 3 - Add Event Listeners

When you open the **index.js** file, you will find the **onDeviceReady** function. This is where the event listener should be added.

```
document.getElementById("createContact").addEventListener("click", createContact);  
document.getElementById("findContact").addEventListener("click", findContact);  
document.getElementById("deleteContact").addEventListener("click", deleteContact);
```

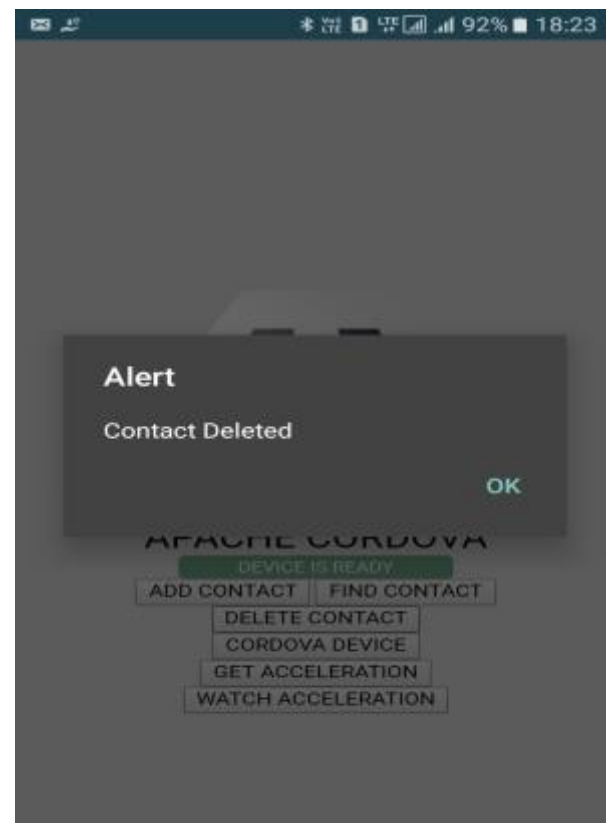
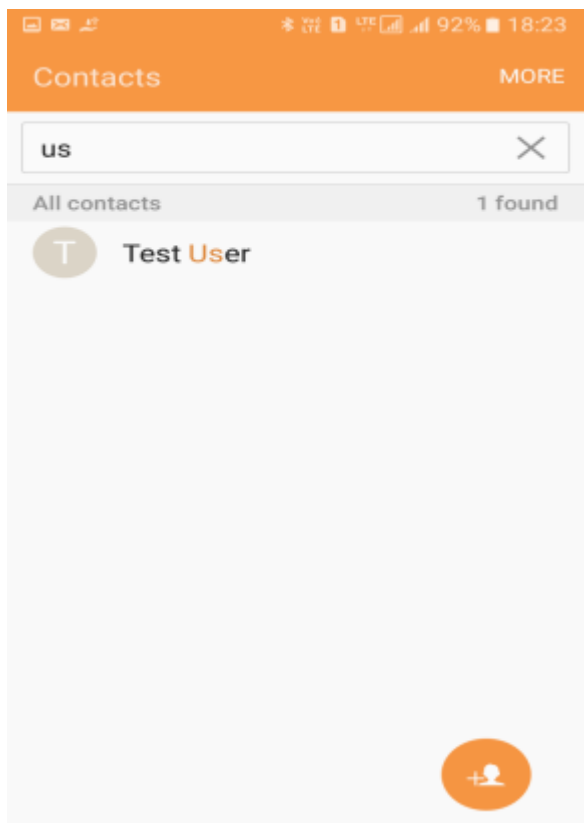
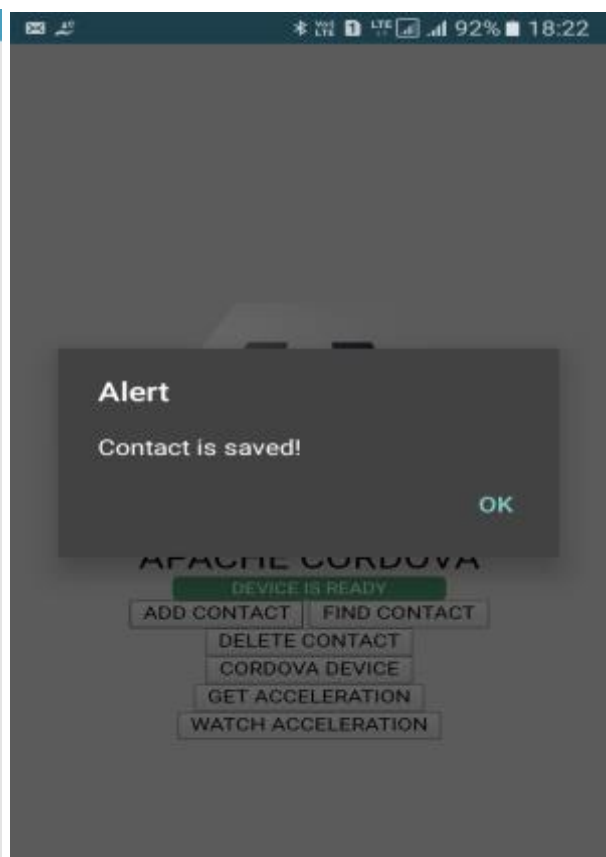
Step 4A- Create Callback Function

```
function createContact() {  
    var myContact = navigator.contacts.create({"displayName": "Test User"});  
    myContact.save(contactSuccess, contactError);  
    function contactSuccess() {  
        alert("Contact is saved!");  
    }  
    function contactError(message) {  
        alert('Failed because: ' + message);  
    }  
}
```

Step 4B- Create Callback Function

```
function deleteContact() {  
    var options = new ContactFindOptions();  
    options.filter = "Test User";  
    options.multiple = false;  
    fields = ["displayName"];  
    navigator.contacts.find(fields, contactfindSuccess, contactfindError, options);  
    function contactfindSuccess(contacts) {  
        var contact = contacts[0];  
        contact.remove(contactRemoveSuccess, contactRemoveError);  
        function contactRemoveSuccess(contact) {  
            alert("Contact Deleted");  
        }  
        function contactRemoveError(message) {  
            alert('Failed because: ' + message);  
        }  
    }  
    function contactfindError(message) {  
        alert('Failed because: ' + message);  
    }  
}
```

OUTPUT



4.2 Installing and using Device plugin

This plugin is used for getting information about the user's device.

Step 1 - Installing Device Plugin

To install this plugin, we need to open the **command prompt** window and run the following code.

```
cordova plugin add cordova-plugin-device
```

Step 2 - Add Buttons

The button will be added. We will place it in the **div class = "app"** in **index.html** file.

```
<button id = "cordovaDevice">CORDOVA DEVICE</button>
```

Step 3 - Add Event Listeners

When you open the **index.js** file, you will find the **onDeviceReady** function. This is where the event listener should be added.

```
document.getElementById("cordovaDevice").addEventListener("click", cordovaDevice);
```

Step 4- Create Callback Function

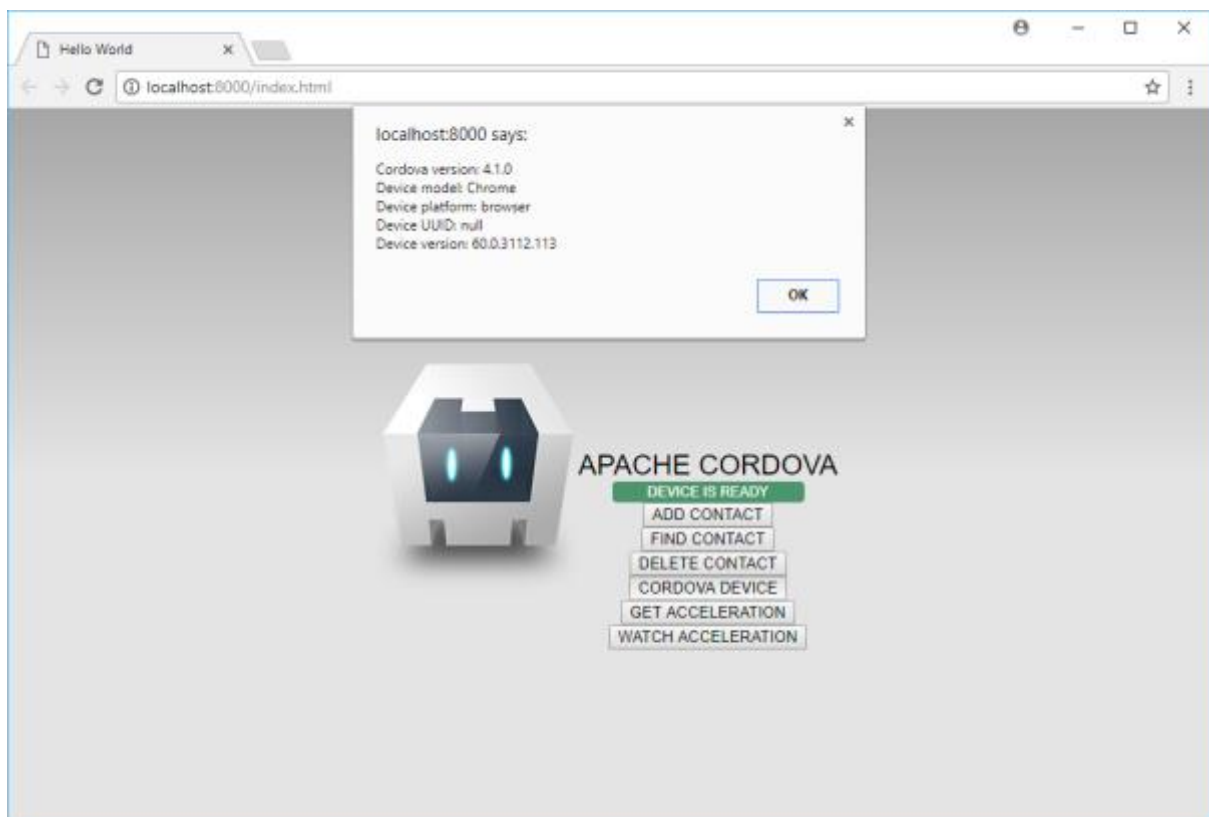
```
function cordovaDevice() {  
    alert("Cordova version: " + device.cordova + "\n" +  
        "Device model: " + device.model + "\n" +  
        "Device platform: " + device.platform + "\n" +  
        "Device UUID: " + device.uuid + "\n" +  
        "Device version: " + device.version);  
}
```

OUTPUT

1. On Real Device (Android Platform)



2. On Browser



4.2 Installing and using Accelerometer plugin

The Accelerometer plugin is also called the **device-motion**. It is used to track device motion in three dimensions.

Step 1 - Installing Accelerometer Plugin

To install this plugin, we need to open the **command prompt** window and run the following code.

```
cordova plugin add cordova-plugin-device-motion
```

Step 2 - Add Buttons

Add two buttons in the **index.html** file.

```
<button id = "getAcceleration">GET ACCELERATION</button>
<button id = "watchAcceleration">WATCH ACCELERATION</button>
```

Step 3 - Add Event Listeners

When you open the **index.js** file, you will find the **onDeviceReady** function. This is where the event listener should be added.

```
document.getElementById("getAcceleration").addEventListener("click",
getAcceleration);

document.getElementById("watchAcceleration").addEventListener(
    "click", watchAcceleration);
```

Step 4- Create Function

```
function getAcceleration() {
    navigator.accelerometer.getCurrentAcceleration(
        accelerometerSuccess, accelerometerError);

    function accelerometerSuccess(acceleration) {
        alert('Acceleration X: ' + acceleration.x + '\n' +
            'Acceleration Y: ' + acceleration.y + '\n' +
            'Acceleration Z: ' + acceleration.z + '\n' +
            'Timestamp: ' + acceleration.timestamp + '\n');
    };
};
```

```

function accelerometerError() {
    alert('onError!');
};
}

function watchAcceleration() {
    var accelerometerOptions = {
        frequency: 3000
    }
    var watchID = navigator.accelerometer.watchAcceleration(
        accelerometerSuccess, accelerometerError, accelerometerOptions);

    function accelerometerSuccess(acceleration) {
        alert('Acceleration X: ' + acceleration.x + '\n' +
            'Acceleration Y: ' + acceleration.y + '\n' +
            'Acceleration Z: ' + acceleration.z + '\n' +
            'Timestamp: ' + acceleration.timestamp + '\n');

        setTimeout(function() {
            navigator.accelerometer.clearWatch(watchID);
        }, 10000);
    };

    function accelerometerError() {
        alert('onError!');
    };
}

```

OUTPUT

