**Please find details about assignment.**

**Solution highlited in GREEN color.**

**Question 1 Solution code:**

//You have text file containing an array of items where some items can be marked as "expired", e.g.

var a = [

{ "name": "item1", "isExpired": false},

{ "name": "item2", "isExpired": true},

{ "name": "item3", "isExpired": true},

{ "name": "item4", "isExpired": false},

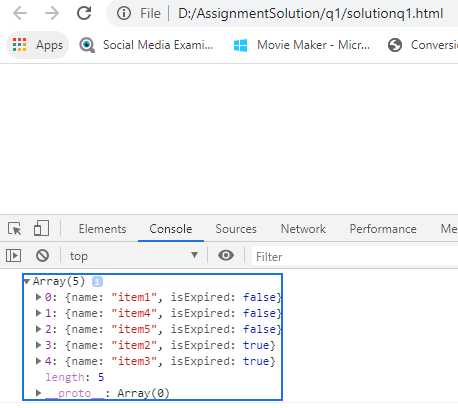
{ "name": "item5", "isExpired": false}

]

a.sort(function(a,b){return a.isExpired-b.isExpired});

alert(JSON.stringify(a));

console.log(a);

Out Put Display in alert box and I put console log as well in code.

**Question 2 Solution code:**

const animal = [

"animals.dogs.poodle",

"animals.cats.tabby",

"animals.cats.siamese",

"animals.dogs.labrador",

"animals.dogs.hound",

"plants.trees",

"animals.birds.parrot.grey"

]

const output = {};

function tree(iterator, outputDisplay) {

let map = outputDisplay;

for(const field of iterator) {

map[field] = map[field] || {};

map = map[field];

}

}

animal.map(field => field.split('.')).forEach(iterator => {

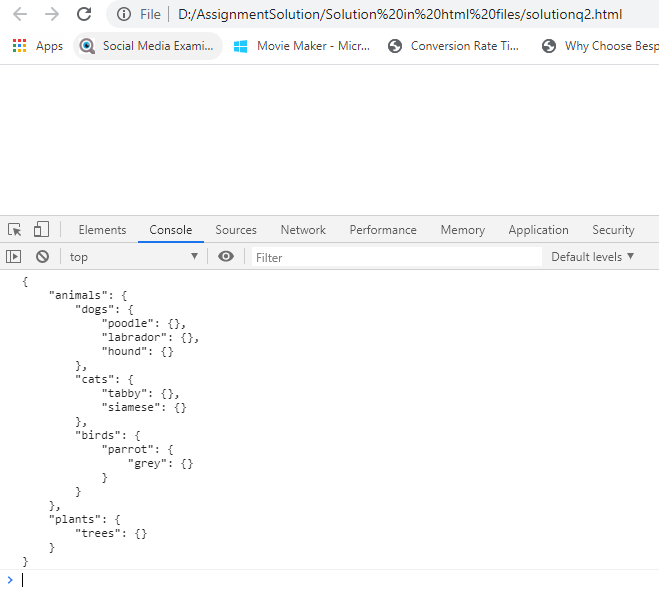
tree(iterator, output);

});

alert(JSON.stringify(output, null, 4));

console.log(JSON.stringify(output, null, 4));

**OUTPUT ON THE BROWSER**



**Question 3 Solution code:**

let obj = {

"animals": {

"dogs": {

"poodle": {},

"labrador": {},

"hound": {}

},

"cats": {

"tabby": {},

"siamese": {}

},

"birds": {

"parrot": {

"grey": {}

}

}

},

"plants": {

"trees": {}

}

};

var store = []

function filePaths(obj, prefix = '') {

for (let key in obj) {

const curPath = `${prefix}/${key}`;

if (typeof obj[key] === 'object') {

store.push(curPath);

filePaths(obj[key], curPath, store);

} else {

store.push(curPath);

}

}

return store;

}

filePaths(obj)

for(var i = 0; i < store.length; i++)

{

for(var j = 0; j < store[i].length; j++)

{

store[i] = store[i].replace('/' , '.');

}

}

function checker(value) {

var search = ['cats'];

for (var i = 0; i < search.length; i++) {

if (value.indexOf(search[i]) > -1) {

return true;

}

}

return false;

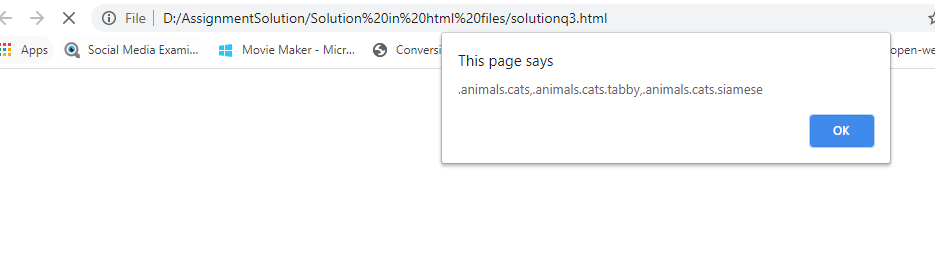
}

store = store.filter(checker);

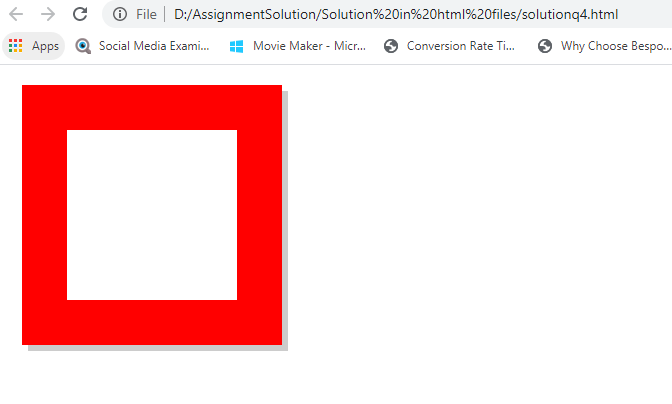
alert(store);

console.log(store);

**OUTPUT ON THE BROWSER**



**Question 4 Solution code:**



**Question 5 Solution code:**

**We have project in angularjs and google released new versions of angular.**

**Architecture wise Angularjs and angular to totally different.**

**I faced challenge to support older version and new version of angular. So created hybrid application.**

**We are downgrading controllers and services and writting code for components.**

**Big challenge updation of technologies and quick learning and supporting application.**

**Installing Angular & ngUpgrade**

**- Removing Bootstrap from index.html**

**<html>**

**<head>**

**<title>Amazing, Inc. Order System</title>**

**</head>**

**<body>**

**<navigation></navigation>**

**<div class="container" ng-view></div>**

**</body>**

**</html>**

**- Changed the AngularJS Module**

**- Created the Angular App Module**

**import { NgModule } from '@angular/core';**

**@NgModule({**

**imports: [**

**BrowserModule,**

**UpgradeModule**

**]**

**})**

**export class AppModule { }**

**- Bootstrapping in the Angular Module**

**- import { NgModule } from '@angular/core';**

**import { BrowserModule } from '@angular/platform-browser';**

**import { UpgradeModule } from '@angular/upgrade/static';**

**import moduleName from './app.module.ajs';**

**@NgModule({**

**imports: [**

**BrowserModule,**

**UpgradeModule**

**]**

**})**

**export class AppModule {**

**constructor(private upgrade: UpgradeModule) { }**

**ngDoBootstrap(){**

**this.upgrade.bootstrap(document.documentElement, [moduleName], {strictDi: true});**

**}**

**}**

**- Creating main.ts**

**- Updating Webpack**

**- Rewriting & Downgrading Component**