**Cordova-project 2APPAI – 2WT - 3IoT**

**Brief description of the app**

The name of the app would be Xapture. An app that would allow users to capture memories together and keep on their device storage. They could make notes as well as poems using the app with special colours to fit their liking.

A person sitting on a chair

Description automatically generated with low confidence.

**Screenshots**

**Graphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generatedA screenshot of a phone

Description automatically generated with medium confidenceA screenshot of a phone

Description automatically generated with medium confidence**

**Used Cordova plug-ins**

**Camera**: I’m using this plugin to take pictures using the device’s camera

**Save Image:** I’m using this plugin store images that have been taken

**Local Notification:** I’m using this plugin to alert the user on waiting times affected by the quotes

**Sqlite:** I’m using this plugin to store notes on the device own database

**Print:** I’m using this plugin to print quotes being called using Ajax methods

**Network Information:** I’m using this plugin to ensure the user is always connected to the internet.

**Infopage about the app**

<https://xapture.sinners.be/>

**Source code**

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport"  
 content="user-scalable=no, initial-scale=1, maximum-scale=1, minimum-scale=1, width=device-width">  
 <link rel="stylesheet" href="https://fonts.googleapis.com/icon?family=Material+Icons"/>  
 <link rel="stylesheet" href="css/index.css">  
 <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/materialize.min.css"/>  
 <title>Xapture</title>  
</head>  
<body>  
<!-- Fixed navbar -->  
<div class="navbar-fixed">  
 <nav class="blue accent-3">  
 <div class="nav-wrapper container">  
 <a href="#!" class="brand-logo center"><img src="img/white-logo-xapture.svg" class="responsive-img logo-top" height="16" alt="Logo"></a>  
 <a href="#!" data-target="nav-mobile" class="sidenav-trigger show-on-large">  
 <i class="material-icons">menu</i>  
 </a>  
 </div>  
 </nav>  
</div>  
<!-- Fixed sideNav -->  
<ul id="nav-mobile" class="sidenav">  
 <li><a href="#!" data-show="tabCamera"><i class="material-icons">camera</i>Camera</a></li>  
 <li>  
 <div class="divider"></div>  
 </li>  
 <li><a href="#!" data-show="tabNotes"><i class="material-icons">notes</i>Notes</a></li>  
 <li>  
 <div class="divider"></div>  
 </li>  
 <li><a href="#!" data-show="tabQuote"><i class="material-icons">light\_mode</i>Quote of the day</a></li>  
 <li>  
 <div class="divider"></div>  
 </li>  
 <li><a href="#!" data-show="tabSetting"><i class="material-icons">settings</i>Setting</a></li>  
 <li>  
 <div class="divider"></div>  
 </li>  
 <li><a href="#!" data-show="tabInfo"><i class="material-icons">info</i>Info</a></li>  
</ul>  
<!-- Grid -->  
<div class="container body ">  
 <div class="row spa" id="tabCamera">  
 <div class="col s12">  
 <h4><i class="material-icons">camera</i> Camera</h4>  
 <img class="materialboxed center-block" id="myImage">  
 <div class="center-align">  
 <button class="center btn save-picture blue accent-3"><i class="material-icons">save</i></button>  
 </div>  
  
 <div class="center camera"></div>  
 <div class="row">  
 <button class="col s6 btn btn-block blue accent-3" id="cameraTakePicture">TAKE PICTURE</button>  
 <button class="col s6 btn blue accent-3" id="cameraGetPicture">GET PICTURE</button>  
 </div>  
  
 <p id="msg"></p>  
 </div>  
 </div>  
  
 <div class="row spa" id="tabNotes">  
 <div class="col s12">  
 <h4><i class="material-icons">notes</i> Notes</h4>  
 <h5 class="disabled center-align grey-text no-notes text-accent-4">You have no notes. Click the button to  
 add notes</h5>  
 <div class="notes row"></div>  
 <a data-target="modal1"  
 class="modal-trigger btn-floating add-note right pin-bottom direction-bottom btn waves-effect blue accent-3 waves-light"><i  
 class="material-icons">add</i></a>  
 <div id="modal1" class="modal" data-colour="">  
 <div class="modal-content">  
 <form action=" ">  
 <label for="modal-header"></label>  
 <input type="text" id="modal-header" class="text-accent-3 center" name="modal-header">  
 <div class="row">  
 <div class="input-field col s12">  
 <textarea id="textarea1" class="materialize-textarea"></textarea>  
 <label for="textarea1">\*-Content-\*</label>  
 </div>  
 </div>  
 </form>  
 <h5 class="black-text center text-accent-1 modal-back">Background colours</h5>  
 <ul class="row colours ">  
 </ul>  
  
 </div>  
 <div class="modal-footer">  
 <a href="#!" class="modal-close save waves-effect waves-green btn-flat">Save</a>  
 </div>  
  
 </div>  
 </div>  
 </div>  
 <div class="row spa" id="tabQuote">  
 <div class="col s12 center">  
 <h4 class="left-align"><i class="material-icons">light\_mode</i> Quote of the Day</h4>  
 <div id="txtHint"></div>  
 <h1 class="ribbon">  
 <strong class="ribbon-content">Your quote would appear here</strong>  
 </h1>  
 <button class="btn waves-effect waves-light center blue accent-3" id="clicky">Random Daily Quote!<i  
 class="material-icons right">send</i></button>  
 <h5 id="demo"></h5>  
 <br>  
 <br>  
 <button class="print btn waves-effect blue accent-3"><i class="material-icons">print</i></button>  
 </div>  
 </div>  
 <div class="row spa" id="tabSetting">  
 <div class="col s12">  
 <h4><i class="material-icons">settings</i>Settings</h4>  
 <h6>Theme</h6>  
 <hr>  
 <div class="switch">  
 <label for="theme">  
 Dark  
 <input type="checkbox" id="theme" name="theme" class="theme" checked/>  
 <span class="lever"></span>  
 Light  
 </label>  
 </div>  
 <br>  
 <!-- <h6>Notifications</h6>-->  
 <!-- <hr>-->  
 </div>  
 </div>  
 <div class="row spa" id="tabInfo">  
 <div class="col s12 container">  
 <h4><i class="material-icons">info</i> Info</h4>  
 <h5 class="center">About Xapture</h5>  
 <div class="divider"></div>  
 <br>  
 <div class="indent">  
 <p>This app is a multipurpose app that helps manage your daily tasks and activities as well as making  
 it pleasurable. You could take notes to help keep track of activities and lists you would work on  
 during your app use. If you encounter errors while using the app, please use the contact form to  
 send a distress message regarding your complaint. Select the type of form from the list that you  
 would like to send to begin. You could also find my personal email and phone  
 number below for more information privately. Explore the app's many functionalities and make sure  
 you have fun!</p>  
 <blockquote><strong>Note: </strong>You need to be connected to the internet to use most of the app's  
 functionalities  
 </blockquote>  
 </div>  
 <br>  
 <h5 class="center">Contact</h5>  
 <div class="divider"></div>  
 <br>  
 <div class="input-field col s12">  
 <select>  
 <option value="" disabled selected>Choose a contact option</option>  
 <option value="complaint">Complaint</option>  
 <option value="suggestions">Suggestions</option>  
 <option value="reviews">Reviews</option>  
 </select>  
 <label>Form Selection</label>  
 </div>  
 <div class="container contact-form">  
 <div class="row center-align">  
 <form class="col s12 comp-form">  
 <div class="row">  
 <div class="input-field col s6">  
 <input placeholder="First Name" name="firstName" id="first\_name" type="text"  
 class="validate">  
 <label for="first\_name">First Name</label>  
 </div>  
 <div class="input-field col s6">  
 <input id="last\_name" placeholder="Last Name" name="lastName" type="text"  
 class="validate">  
 <label for="last\_name">Last Name</label>  
 </div>  
 </div>  
 <div class="row">  
 <div class="input-field col s12">  
 <input id="email" type="email" name="email" class="validate">  
 <label for="email">Email</label>  
 <span class="helper-text" data-error="inappropriate format" data-success="Good to go">Validator</span>  
 </div>  
 </div>  
 <div class="row">  
 <form class="col s12">  
 <div class="row">  
 <div class="input-field col s12">  
 <textarea id="complaint" name="complaint"  
 class="materialize-textarea"></textarea>  
 <label for="complaint" class="complaint-label">Form</label>  
 </div>  
 </div>  
 <label for="type"></label>  
 <input type="hidden" id="type" name="type">  
 </form>  
 </div>  
 <input type="submit" placeholder="Submit" value="submit" class="btn comp blue accent-3">  
 </form>  
  
 </div>  
  
 </div>  
 <div class="wrapper confirm-comp">  
 <svg class="checkmark" xmlns="http://www.w3.org/2000/svg" viewBox="0 0 52 52">  
 <circle class="checkmark\_\_circle" cx="26" cy="26" r="25" fill="none"/>  
 <path class="checkmark\_\_check" fill="none" d="M14.1 27.2l7.1 7.2 16.7-16.8"/>  
 </svg>  
 <h6 class="center-align">Sent!</h6>  
 </div>  
 <div class="w4rAnimated\_checkmark">  
 <svg version="1.1" xmlns="http://www.w3.org/2000/svg" viewBox="0 0 130.2 130.2">  
 <circle class="path circle" fill="none" stroke="#D06079" stroke-width="6" stroke-miterlimit="10"  
 cx="65.1" cy="65.1" r="62.1"/>  
 <line class="path line" fill="none" stroke="#D06079" stroke-width="6" stroke-linecap="round"  
 stroke-miterlimit="10" x1="34.4" y1="37.9" x2="95.8" y2="92.3"/>  
 <line class="path line" fill="none" stroke="#D06079" stroke-width="6" stroke-linecap="round"  
 stroke-miterlimit="10" x1="95.8" y1="38" x2="34.4" y2="92.2"/>  
 </svg>  
 <h6 class="center-align">Contact provider privately regarding your problem</h6>  
 </div>  
 <h5 class="center">Personal Contact</h5>  
 <div class="divider"></div>  
  
 <table class="center responsive-table highlight">  
 <thead>  
 <tr>  
 <th>Media</th>  
 <th>Media Platform</th>  
 <th>Media Link</th>  
 </tr>  
 </thead>  
  
 <tbody>  
 <tr>  
 <td>  
 <i class="material-icons">call</i>  
 </td>  
 <td>Phone</td>  
 <td><a href="tel:0497190731">0497190731</a></td>  
 </tr>  
 <tr>  
 <td>  
 <i class="material-icons">textsms</i>  
 </td>  
 <td>SMS</td>  
 <td><a href="sms:0497190731">Click to text me</a></td>  
 </tr>  
 <tr>  
 <td>  
 <i class="material-icons">email</i>  
 </td>  
 <td>Email</td>  
 <td><a href="mailto:r0855536@student.thomasmore.be">r0855536@student.thomasmore.be</a></td>  
 </tr>  
 </tbody>  
 </table>  
 <br>  
 </div>  
 </div>  
 <div class="row spa" id="tabNoteInternet">  
 <div class="col s12">  
 <h4 class="center-align"><i class="material-icons">wifi\_off</i> No Internet Connection</h4>  
 <div>  
 <video src="img/animated\_medium20190830-29222-m6anl7%20(1).mp4" type="video/mp4" class="wifi" autoplay  
 loop></video>  
 </div>  
 </div>  
 </div>  
  
</div>  
  
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>  
<script src="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/js/materialize.min.js"></script>  
<script src="cordova.js"></script>  
<script src="js/theme.js"></script>  
<script src="js/camera.js"></script>  
<script src="js/form.js"></script>  
<script src="js/database.js"></script>  
  
<script defer src="js/app.js"></script>  
<script>  
 $(***document***).ready(function () {  
 if (/Android/i.test(***navigator***.userAgent)) {  
 delete ***window***.sqlitePlugin;  
 }  
 $('.ribbon-content').addClass('grey-text')  
  
 let count = 0;  
 if (***localStorage***.getItem('count')) {  
 $('.ribbon-content').text(***localStorage***.getItem("quote")).removeClass('grey-text');  
 count = parseInt(***localStorage***.getItem('count'));  
 if (count === 4) {  
 $("#demo").empty();  
 $("#clicky").addClass('disabled');  
 let nextDay = parseInt(***localStorage***.getItem('next'))  
 var x = setInterval(function () {  
 // Get today's date and time  
 var now = new ***Date***().getTime();  
 // Find the distance between now and the count down date  
 var distance = nextDay - now;  
 // Time calculations for days, hours, minutes and seconds  
 var days = ***Math***.floor(distance / (1000 \* 60 \* 60 \* 24));  
 var hours = ***Math***.floor((distance % (1000 \* 60 \* 60 \* 24)) / (1000 \* 60 \* 60));  
 var minutes = ***Math***.floor((distance % (1000 \* 60 \* 60)) / (1000 \* 60));  
 var seconds = ***Math***.floor((distance % (1000 \* 60)) / 1000);  
 $("#demo").show();  
 // Output the result in an element with id="demo"  
 $("#demo").text("Waiting for: " + hours + "h "  
 + minutes + "m " + seconds + "s ");  
  
 // If the count down is over, write some text  
 if (distance < 0) {  
 clearInterval(x);  
 $('.ribbon .ribbon-content:before, .ribbon .ribbon-content:after').css({  
 "background": "#986794",  
 'border': '1.5em solid #986794',  
 });  
 $('.ribbon:before, .ribbon:after ').css({  
 'background': '#986794',  
 'border': "1.5em solid #986794"  
 })  
 $('.ribbon').css({  
 'background': '#986794',  
 })  
 $("#demo").text("Countdown Expired");  
 $("#demo").show().delay(5000).fadeOut();  
 $('.ribbon-content').text('Your quote would appear here').addClass("grey-text");  
 cordova.plugins.notification.local.schedule({  
 title: "Time's up!!",  
 text: "Time's up to receive your daily quote, don't forget to hydrate!",  
 foreground: true  
 });  
 // localStorage.setItem('count', 0);  
 ***localStorage***.removeItem("count");  
 $("#clicky").removeClass('disabled');  
 count = 0;  
 $('#clicky').on('click', function () {  
 count += 1  
 ***localStorage***.setItem('count', count);  
 //console.log(count)  
 getRandomQuote();  
  
 });  
 }  
 }, 1000);  
 $('.ribbon .ribbon-content:before, .ribbon .ribbon-content:after').css({  
 "background": "red",  
 'border': '1.5em solid red',  
 });  
 $('.ribbon:before, .ribbon:after ').css({  
 'background': 'red',  
 'border': "1.5em solid red"  
 })  
 $('.ribbon').css({  
 'background': 'red',  
 })  
  
 } else {  
 $('#clicky').on('click', function () {  
 count += 1  
 ***localStorage***.setItem('count', count);  
 //console.log(count)  
 getRandomQuote();  
  
 });  
 }  
  
 } else {  
 $('#clicky').on('click', function () {  
 count += 1  
 ***localStorage***.setItem('count', count);  
 getRandomQuote();  
  
 });  
 }  
  
 function getRandomQuote() {  
 if (***localStorage***.getItem('count') < 4) {  
 let text = 1;  
 $.ajax({ //create an ajax request to display.php  
 type: "POST",  
 data: {  
 text: text  
 },  
 url: "https://nathan655.sinners.be/ajax.php",  
 dataType: "json", //expect html to be returned  
 success: function (response) {  
 function randomNumber(min, max) {  
 return ***Math***.floor(***Math***.random() \* (max - min)) + min;  
 }  
  
 const num = randomNumber(0, response.length);  
 $('.ribbon-content').text(response[num][0]+"\n and you're probably feeling "+response[num][1])  
 .removeClass('grey-text')  
 ***localStorage***.setItem('quote', response[num][0])  
 $('.print').show().on('click', function () {  
 // cordova.plugins.printer.print(`${response[num][1]}`);  
 cordova.plugins.printer.print();  
 });  
 }  
  
 });  
 } else {  
 $("#clicky").addClass('disabled');  
 const now = new ***Date***().getTime();  
 alert('You can only get messages three times a day');  
 $("#demo").empty();  
 $("#demo").show();  
 var countDownDate = new ***Date***(now + 86400000).getTime();  
 ***localStorage***.setItem('next', countDownDate);  
 var x = setInterval(function () {  
 let nextDay = ***localStorage***.getItem('next');  
  
 // Get today's date and time  
 var now = new ***Date***().getTime();  
  
 // Find the distance between now and the count down date  
 var distance = nextDay - now;  
  
 // Time calculations for days, hours, minutes and seconds  
 var days = ***Math***.floor(distance / (1000 \* 60 \* 60 \* 24));  
 var hours = ***Math***.floor((distance % (1000 \* 60 \* 60 \* 24)) / (1000 \* 60 \* 60));  
 var minutes = ***Math***.floor((distance % (1000 \* 60 \* 60)) / (1000 \* 60));  
 var seconds = ***Math***.floor((distance % (1000 \* 60)) / 1000);  
  
 // Output the result in an element with id="demo"  
 $("#demo").text("Waiting for: " + hours + "h "  
 + minutes + "m " + seconds + "s ");  
  
 // If the count down is over, write some text  
 if (distance < 0) {  
 clearInterval(x);  
 $('.ribbon .ribbon-content:before, .ribbon .ribbon-content:after').css({  
 "background": "#986794",  
 'border': '1.5em solid #986794',  
 });  
 $('.ribbon:before, .ribbon:after ').css({  
 'background': '#986794',  
 'border': "1.5em solid #986794"  
 })  
 $('.ribbon').css({  
 'background': '#986794',  
 })  
 $("#demo").text("Countdown Expired");  
 $("#demo").show().delay(5000).fadeOut();  
 $('.ribbon-content').text('Your quote would appear here').addClass("grey-text");  
  
 cordova.plugins.notification.local.schedule({  
 title: "Time's up!!",  
 text: "Time's up to receive your daily quote, don't forget to hydrate!",  
 foreground: true  
 });  
 ***localStorage***.setItem('count', 0);  
 count = 0;  
 $("#clicky").removeClass('disabled');  
 }  
 }, 1000);  
 $('.ribbon .ribbon-content:before, .ribbon .ribbon-content:after').css({  
 "background": "red",  
 'border': '1.5em solid red',  
 });  
 $('.ribbon:before, .ribbon:after ').css({  
 'background': 'red',  
 'border': "1.5em solid red"  
 })  
 $('.ribbon').css({  
 'background': 'red',  
 })  
  
 }  
 }  
  
 }, false);  
</script>  
</body>  
</html>

$(function () {  
 ***document***.addEventListener("deviceready", onDeviceReady, false);  
 $('#tabCamera').show();  
 $('.print').hide();  
 $('#tabInfo').hide();  
 $("#demo").hide();  
 $('#tabNotes').hide();  
 $('#tabQuote').hide();  
 $('#tabSetting').hide();  
 $('#tabNoteInternet').hide();  
 $('.save-picture').hide();  
 $('.contact-form').hide();  
 $('.confirm-comp').hide();  
 $('.w4rAnimated\_checkmark').hide();  
 //console.log($('.wifi'))  
 setInterval(() => {  
 if (***navigator***.connection.type === 'none') {  
 $('.print').hide();  
 $('#tabInfo').hide();  
 $("#demo").hide();  
 $('#tabNotes').hide();  
 $('#tabQuote').hide();  
 $('#tabSetting').hide();  
 $('#tabNoteInternet').show();  
 $('#tabCamera').hide();  
 } else {  
 $('#tabNoteInternet').hide();  
 }  
 }, 5000);  
  
 $('.sidenav').sidenav(); /\* https://materializecss.com/sidenav.html \*/  
  
 $('.sidenav a').click(function () {  
 $('.spa').hide();  
 $('#' + $(this).data('show')).show();  
 $('.sidenav').sidenav('close');  
 });  
 $('select').formSelect();  
  
  
});  
  
function onDeviceReady() {  
 //console.log('Running cordova-' + cordova.platformId + '@' + cordova.version);  
 $('.modal').modal();  
 $('.materialboxed').materialbox();  
 // Camera  
 $('#cameraTakePicture').on('click', function () {  
 ***Camera***.takePicture();  
 })  
 $('#cameraGetPicture').on('click', function () {  
 ***Camera***.getPicture();  
 })  
 $('.save-picture').on('click', function () {  
 ***Camera***.savePicture();  
 })  
 // Database  
 ***Sqlite***.initDatabase();  
 $('.notes').on('click', '.delete', function () {  
 ***Sqlite***.id($(this).data('id'));  
 ***Sqlite***.delete()  
 })  
 .on('click', '.edit', function (e) {  
 ***Sqlite***.id($(this).data('id'));  
 ***Sqlite***.edit(e);  
 $('.save-edit').on('click', function (e) {  
 ***Sqlite***.saveEdit(e)  
 })  
 })  
 $('.save').on('click', function () {  
 ***Sqlite***.save();  
 });  
  
 // Form  
  
 // Theme  
 if (***localStorage***.getItem("theme")) {  
 ***Theme***.changeTheme();  
 $('#theme').on('click', function () {  
 ***Theme***.checkTheme();  
 ***Theme***.changeTheme();  
 })  
 } else {  
 ***localStorage***.setItem('theme', 'light');  
 // $('#theme')[0].checked( true);  
 }  
}

const ***Sqlite*** = function () {  
 let count = 0;  
 let caseTableResult = [];  
 let total = 0;  
 let id;  
 let previousColour = "";  
 let DB;  
 let runiOS = false;  
 const colours = {  
 red: "red",  
 yellow: "yellow",  
 blue: 'light-blue',  
 teal: 'teal',  
 purple: "purple",  
 lightGreen: "light-green"  
 }  
 $('.note-card').on('drag', function () {  
 //console.log($(this).position())  
 })  
 //console.log($('.note-card'))  
 $('.add-note').click(function () {  
 $('#modal-header').val("")  
 $('#textarea1').val("");  
 id = 0;  
 $('#modal1').removeClass($('#modal1').data('colour'))  
 $('.modal-footer').removeClass($('#modal1').data('colour'))  
 $('.modal-back').removeClass('white-text');  
 $('.save-edit').addClass('save')  
 $('.save').removeClass('white-text save-edit')  
 $('.materialize-textarea').removeClass('white-text')  
 $('#modal-header').removeClass('white-text')  
 $('#modal1').data('colour', "")  
  
 })  
  
 $.each(colours, function (id, name) {  
 $('.colours').append(`<li class="col s2"><a href="#!" class="change-modal btn-floating ${name} small waves-effect waves-circle" data-colour="${name}"></a></li>`)  
  
 })  
 let notYellow = function () {  
 if ($("#modal1").data('colour') === "yellow") {  
 $('.modal-back').removeClass('white-text');  
 $('.save').removeClass('white-text')  
 $('.materialize-textarea').removeClass('white-text')  
 $('#modal-header').removeClass('white-text')  
 } else if ($("#modal1").data('colour') === "") {  
 $('.modal-back').removeClass('white-text');  
 $('.save').removeClass('white-text')  
 $('.materialize-textarea').removeClass('white-text')  
 $('#modal-header').removeClass('white-text')  
 } else {  
 $('.modal-back').addClass('white-text');  
 $('.save').addClass('white-text')  
 $('.materialize-textarea').addClass('white-text')  
 $('#modal-header').addClass('white-text')  
 }  
 }  
 let removePrevious = function () {  
 $('#modal1').removeClass(previousColour);  
 $('.modal-footer').removeClass(previousColour);  
 }  
 $('.change-modal').on('click', function () {  
 $('#modal1').data('colour', $(this).data('colour'));  
 notYellow();  
 removePrevious();  
 previousColour = $(this).data('colour');  
 $('.modal-footer').addClass($('#modal1').data('colour'));  
 $('#modal1').addClass($('#modal1').data('colour'));  
 })  
  
 function initDatabase() {  
 const onDeviceReady = new Promise(function (resolve, reject) {  
 if (***document***.URL.match(/^https?:/i)) {  
 //console.log("Running in a browser...");  
 resolve();  
 } else {  
 //console.log("Running in an app...");  
 runiOS = true;  
 ***document***.addEventListener("deviceready", resolve, false);  
 }  
 });  
 if (runiOS) {  
 DB = ***window***.sqlitePlugin.openDatabase({name: 'my.db', location: 'default'}, function (db) {  
 }, function (error) {  
 //console.log('Open database ERROR: ' + JSON.stringify(error));  
 });  
 DB.transaction(function (tx) {  
 tx.executeSql('CREATE TABLE IF NOT EXISTS DemoTable (id INTEGER PRIMARY KEY AUTOINCREMENT,head TEXT, content TEXT, colour varchar)');  
 }, function (error) {  
 //console.log('Transaction ERROR: ' + error.message);  
 }, function () {  
 //console.log('Populated database OK');  
 });  
 //console.log('DB: SQLite');  
 } else {  
 DB = ***window***.openDatabase('my', "0.1", "My list", 200000);  
 //console.log('DB: WebSQL');  
 DB.transaction(function (tx) {  
 tx.executeSql('CREATE TABLE IF NOT EXISTS DemoTable (id INTEGER PRIMARY KEY AUTOINCREMENT,head TEXT, content TEXT, colour varchar)');  
 tx.executeSql('SELECT \* FROM DemoTable', [], querySuccess, errorCB);  
 }, function (error) {  
 //console.log('Transaction ERROR: ' + error.message);  
 }, function () {  
 //console.log('Populated database OK');  
 });  
 }  
  
  
 }  
  
 function errorCB(err) {  
 alert("Error processing SQL: " + err.code);  
 }  
  
 let querySuccess = function (tx, result) {  
 $('.notes').empty();  
 caseTableResult = result.rows;  
 // const keys =Object.keys(caseTableResult).reverse();  
 // //console.log(keys)  
 let total = caseTableResult.length;  
 if (total === 0) {  
 $('.no-notes').show('slide')  
 } else {  
 $('.no-notes').hide("slide")  
 }  
 if (count === caseTableResult.length) {  
 count = 0  
 $('#folderData').empty();  
 } else {  
 $.each(caseTableResult, function (id, data) {  
 let color = data.colour  
 //console.log(data)  
 if (color === '') {  
 color = "blue-grey";  
 }  
 if (data.head === "" && data.content === "") {  
 DB.transaction(function (tx) {  
 tx.executeSql(`DELETE  
 FROM DemoTable  
 where id = ${data.id}`);  
 }, function (error) {  
 //console.log('Transaction ERROR: ' + error.message);  
 }, function () {  
 //console.log('Populated database OK');  
 });  
  
 } else {  
 notYellow();  
 $('.notes').append(`<div class="col s12 row note-card" id="data-${data.id}">  
 <div class="col s12 m6 ">  
 <div class="card ${color}" data-colour="${color}">  
 <div class="card-content database ">  
 <span class="card-title">${data.head}</span>  
  
 <p class="card-content">${data.content}</p>  
 </div>  
 <a data-id="${data.id}" class="btn-floating halfway-fab waves-effect right waves-light delete red"><i class="material-icons">delete\_forever</i></a>  
 <a data-id="${data.id}" class="btn-floating halfway-fab waves-effect left waves-light edit green"><i class="material-icons">edit</i></a>  
 </div>  
 </div>  
 </div>`)  
 }  
  
 })  
 }  
 }  
 let editNote = function (e) {  
 $('.save').addClass('save-edit')  
 $('.save-edit').removeClass('save')  
 console.log(id);  
 e.stopImmediatePropagation();  
 DB.transaction(function (tx) {  
 tx.executeSql(`SELECT \*  
 FROM DemoTable  
 where id = ${id}`, [], populateModal, errorCB);  
 }, function (error) {  
 console.log('Transaction ERROR: ' + error.message);  
 }, function () {  
 // Success Callback  
 });  
  
 let populateModal = function (tx, result) {  
 let response = result.rows[0];  
 console.log(response);  
 $('#modal-header').val(response.head);  
 $('#textarea1').val(response.content);  
  
 previousColour = $('#modal1').data('colour');  
 removePrevious();  
 $('#modal1').data('colour', response.colour);  
 //console.log($('#modal1').data('colour'))  
 $('#modal1').addClass($('#modal1').data('colour'));  
 $('.modal-footer').addClass($('#modal1').data('colour'));  
 notYellow();  
 const instance = M.Modal.getInstance($('#modal1'))  
 instance.open();  
 }  
  
 }  
 let intId = function(int){  
 id =int;  
 }  
  
 function updateQuery(tx) {  
 const head = $('#modal-header').val();  
 const content = $('#textarea1').val();  
 let colour = $('#modal1').data('colour');  
 // Insert query  
 tx.executeSql('INSERT INTO DemoTable(head, content, colour) VALUES (?,?,?)', [head, content, colour], querySuccess, errorCB);  
 tx.executeSql(`DELETE  
 FROM DemoTable  
 where id = ${id}`);  
 }  
  
 let saveEdit = function (e) {  
 e.stopImmediatePropagation();  
 M.toast({html: "Data Updated Successfully !", classes: ' blue '});  
 const instance = M.Modal.getInstance($('#modal1'));  
 instance.close();  
 DB.transaction(updateQuery, errorCB);  
 //console.log(id)  
 DB.transaction(function (tx) {  
 tx.executeSql('SELECT \* FROM DemoTable', [], querySuccess, errorCB);  
 }, function (error) {  
 //console.log('Transaction ERROR: ' + error.message);  
 }, function () {  
 //console.log('Populated database OK');  
 })  
 $('#modal-header').val("")  
 $('#textarea1').val("");  
 }  
  
 let loadNotes = function () {  
 DB.transaction(function (tx) {  
 tx.executeSql('SELECT \* FROM DemoTable', [], querySuccess, errorCB);  
 }, function (error) {  
 //console.log('Transaction ERROR: ' + error.message);  
 }, function () {  
 //console.log('Populated database OK');  
 });  
 }  
 let saveNote = function () {  
 let title = $('#modal-header').val();  
 let content = $('#textarea1').val();  
 let colour = $('#modal1').data('colour');  
 if (title === "" && content === "") {  
 M.toast({html: "You didn't put anything to your database", classes: 'red'})  
 } else {  
  
 DB.transaction(function (tx) {  
 tx.executeSql('INSERT INTO DemoTable(head, content, colour) VALUES (?,?,?)', [title, content, colour]);  
 tx.executeSql('SELECT \* FROM DemoTable', [], querySuccess, errorCB);  
 }, function (error) {  
 //console.log('Transaction ERROR: ' + error.message);  
 }, function () {  
 //console.log('Populated database OK');  
 });  
  
 $('#modal-header').val("")  
 $('#textarea1').val("");  
 }  
 }  
 let deleteNote = function () {  
 if (confirm("Are you sure you want to delete this note") === true) {  
 M.toast({html: 'Deleting this Note'});  
 setTimeout(  
 function () {  
 $(`#data-${id}`).fadeOut("slow");  
 DB.transaction(function (tx) {  
 tx.executeSql(`DELETE  
 FROM DemoTable  
 where id = ${id}`);  
 }, function (error) {  
 //console.log('Transaction ERROR: ' + error.message);  
 }, function () {  
 // Success Callback  
 });  
  
 loadNotes();  
 }, 2000);  
 } else {  
 // Don't delete  
 }  
 }  
  
 return {  
 initDatabase: initDatabase,  
 delete: deleteNote,  
 edit: editNote,  
 save: saveNote,  
 saveEdit: saveEdit,  
 id:intId,  
 }  
}();

$('select').change(function () {  
 //console.log($(this).val())  
 let text = capitalizeFirstLetter($(this).val())  
 $('.complaint-label').text('Form ' + text)  
 $('#type').val(text)  
 $('.contact-form').show();  
})  
$('.comp').on('click', function () {  
 //console.log($('#type').val())  
 $.ajax({  
 url: 'https://nathan655.sinners.be/form.php',  
 type: 'POST',  
 data: {  
 type:$('#type').val(),  
 firstName: $('#first\_name').val(),  
 lastName:$('#last\_name').val(),  
 email: $('#email').val(),  
 complaint: $('#complaint').val()  
 },  
 success: function(msg) {  
 alert('Email Sent');  
 if (msg==="Connected Successfully.New complaint created successfully"){  
 $('.contact-form').hide("slideUp").delay(2000);  
 $('.confirm-comp').show().delay(2000).fadeOut();  
 }else {  
 $('.w4rAnimated\_checkmark').show().delay(5000).fadeOut();  
 }  
 //console.log(msg);  
 }  
 });  
});  
  
  
function capitalizeFirstLetter(string) {  
 return string.charAt(0).toUpperCase() + string.slice(1);  
}  
const form = function (){  
  
}

let ***currentTheme*** = ***localStorage***.getItem('theme');  
root = ***document***.documentElement;  
let light = function () {  
 $('body').removeClass('blue-grey darken-3 white-text')  
 $('#nav-mobile').removeClass('blue-grey darken-3 white-text')  
 $('#nav-mobile a').removeClass('white-text');  
 $('#nav-mobile i').removeClass('white-text');  
 $('input').removeClass('white-text');  
 $('textarea').removeClass('white-text');  
 $('td a').removeClass('white-text').addClass('black-text');  
}  
  
let ***Theme*** = (function () {  
 let dark = function () {  
 $('body').addClass('blue-grey darken-3 white-text');  
 $('#nav-mobile').addClass('blue-grey darken-3 white-text');  
 $('#nav-mobile a').addClass('white-text');  
 $('#nav-mobile i').addClass('white-text');  
 $('input').addClass('white-text');  
 $('textarea').addClass('white-text');  
 $('td a').addClass('white-text');  
 }  
 const checkTheme = function () {  
 if ($('#theme')[0].checked) {  
 //console.log('checked')  
 ***localStorage***.setItem('theme', 'light');  
 ***currentTheme*** = ***localStorage***.getItem('theme');  
 //console.log(currentTheme);  
 } else {  
 //console.log('not checked')  
 ***localStorage***.setItem('theme', 'dark');  
 ***currentTheme*** = ***localStorage***.getItem('theme');  
 //console.log(currentTheme);  
 }  
 }  
 let changeTheme = function () {  
 if (***currentTheme*** === 'dark') {  
 $('#theme').attr('checked', false);  
 root.setAttribute('data-theme', ***currentTheme***);  
 dark();  
 } else {  
 root.setAttribute('data-theme', ***currentTheme***);  
 light();  
 }  
 }  
  
  
 return {  
 checkTheme: checkTheme,  
 changeTheme: changeTheme  
 }  
})();

let ***Camera*** = function () {  
 function cameraTakePicture() {  
  
 function onSuccess(imageData) {  
 $('#myImage').attr('src', "data:image/jpeg;base64," + imageData);  
 //console.log(image.src);  
 $('.save-picture').show();  
 }  
  
 function onFail(message) {  
 alert('Failed because: ' + message);  
 }  
  
 ***navigator***.camera.getPicture(onSuccess, onFail, {  
 quality: 50,  
 // destinationType: Camera.DestinationType.FILE\_URI  
 });  
 }  
  
 function cameraSavePicture() {  
 let image = $('#myImage').attr('src');  
 ***window***.cordova.plugins.imagesaver.saveImageToGallery(image, onSaveImageSuccess, onSaveImageError);  
  
 function onSaveImageSuccess(e) {  
 //console.log('--------------success');  
 }  
  
 function onSaveImageError(error) {  
 //console.log('--------------error: ' + error);  
 }  
 }  
  
 function cameraGetPicture() {  
 ***navigator***.camera.getPicture(onSuccess, onFail, {  
 quality: 70,  
 destinationType: ***Camera***.DestinationType.FILE\_URI,  
 sourceType: ***Camera***.PictureSourceType.PHOTOLIBRARY  
 });  
  
 function onSuccess(imageURL) {  
 $('#myImage').attr('src', imageURL);  
 }  
  
 function onFail(message) {  
 alert('Failed because: ' + message);  
 }  
  
 }  
  
 return {  
 takePicture: cameraTakePicture,  
 getPicture: cameraGetPicture,  
 savePicture: cameraSavePicture,  
 }  
}();