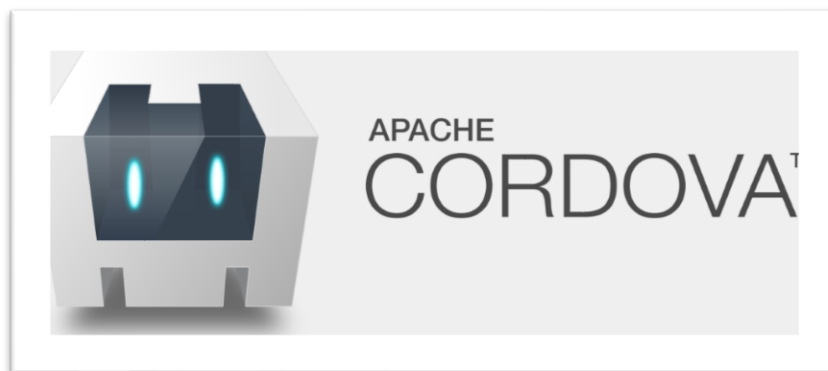
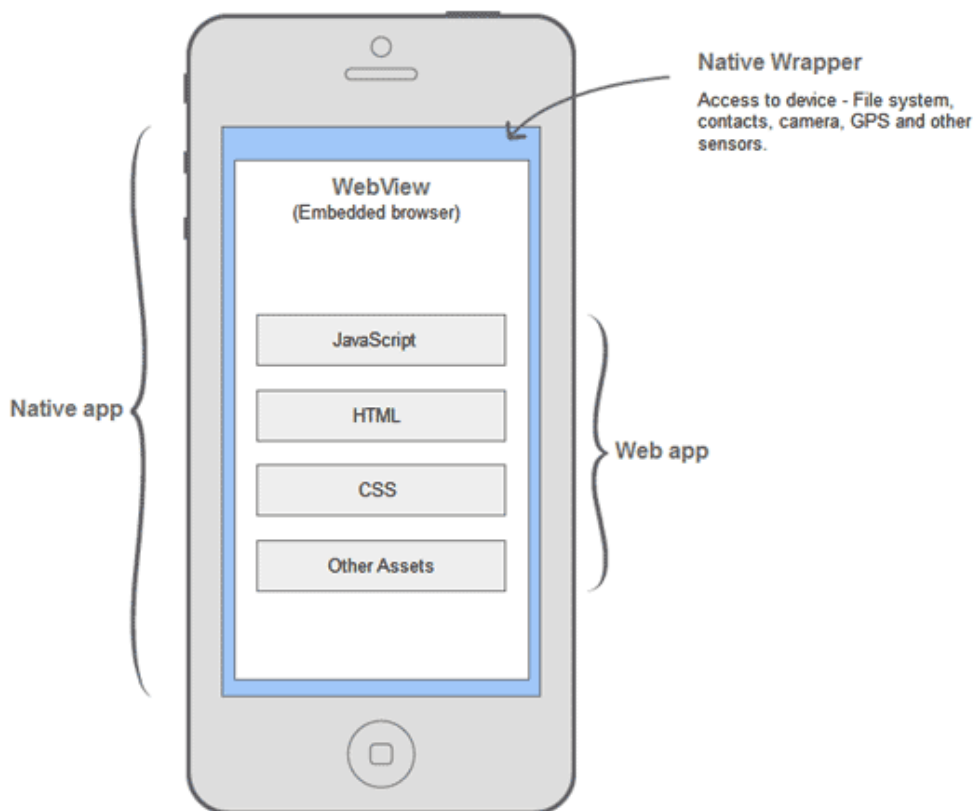


Въведение с NPM, Apache Cordova и jQuery Mobile



Apache Cordova е рамка за разработване на мобилни приложения, създадена от Nitobi. Adobe Systems купиват Nitobi през 2011 г., преименуват го на PhoneGap и по-късно пуснат версия с отворен код на софтуера, наречена Apache Cordova. Тя дава възможност на софтуерните програмисти да създават хибридни уеб приложения за мобилни устройства, използвайки CSS3, HTML5 и JavaScript, вместо да разчитат на специфични за платформа API, като тези в Android, iOS или Windows Phone. Получените приложения са хибридни, което означава, че те не са нито истински мобилно приложение (защото цялото изобразяване на оформлението се извършва чрез уеб изгледи вместо вградената рамка на потребителския интерфейс на платформата), нито чисто уеб базирани (тъй като те не са само уеб приложения, а са пакетирани като приложения за разпространение и имат достъп до API на собственото устройство). Приложенията, които изграждате с кордова, се изпълняват в родните приложения (native app), наречени обвивки (wrappers), насочени към всяка мобилна платформа.



1. Изисквания

Моля, инсталирайте Node JS от официалния сайт: <https://nodejs.org/en/> .

2. Първи стъпки

1. Инсталиране на Cordova

Командният ред на Cordova работи на Node.js и е достъпен чрез NPM. Отворете терминал и въведете *npm install -g cordova*.

2. Създайте проект

Създайте празен проект на Cordova с помощта на команден ред. Придвижете се до директорията, където искате да създадете своя проект и напишете:

```
cordova create StudentAdmin
```

За пълен набор от опции напишете *cordova help create*.

3. Добавете платформа

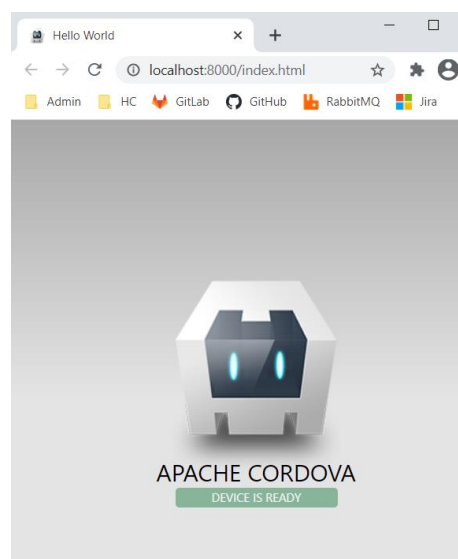
След като създадете проект в Cordova, отидете в директорията на проекта. От директорията на проекта трябва да добавите платформа, за която искате да изградите приложението си. Като за начало изпълнете следните команди:

```
- cd StudentAdmin
```

```
- cordova platform add browser
```

4. Стартирайте приложението си

От командния ред стартирайте *cordova run browser*



3. jQuery Mobile



оптимизирана за докосване уеб рамка (известна също като мобилна рамка), по-точно библиотека на JavaScript, разработена от екипа на проекта jQuery. Разработката се фокусира върху създаването на рамка, съвместима с голямо разнообразие от смартфони и таблети.

Характеристики:

- Съвместим с всички основни браузъри за настолни компютри, както и с всички основни мобилни платформи, включително Android, iOS, Windows Phone, Blackberry, WebOS, Symbian.
- Построен върху ядрото на jQuery
- Позволява създаването на персонализирани теми <https://themroller.jquerymobile.com/>.
- Навигация с AJAX с анимирани преходи на страници, която осигурява възможност за създаване на семантични URL адреси чрез pushState.

В директорията на приложението изпълнете следната команда:

- `npm install -g cordova-jquery`

А след това, в същата директория изпълнете само:

- `cordova-jquery`

В конзолата ще се появят няколко въпроса, моля отговорете по следния начин:

? Would you like to add jQuery mobile to the current Apache Cordova project? **Yes (Y)**

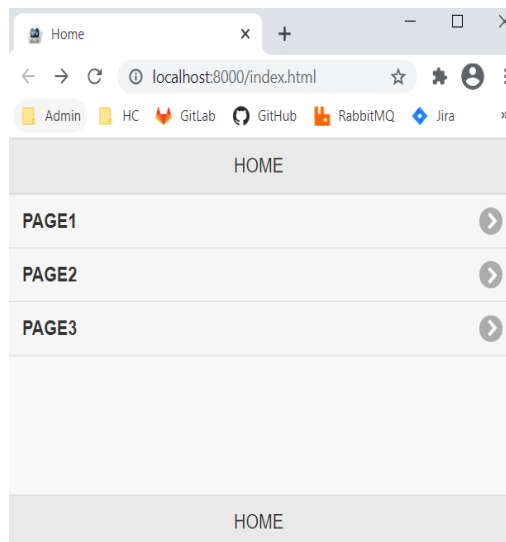
? What would you like to do now that jQuery is enabled? **applyTemplate**

? Which jQuery mobile template would you like to apply to your Apache Cordova project? **listView**

? Would you like to keep the current code? ...it could get ugly! **No (n)**

След изпълнение на командата `cordova run browser`

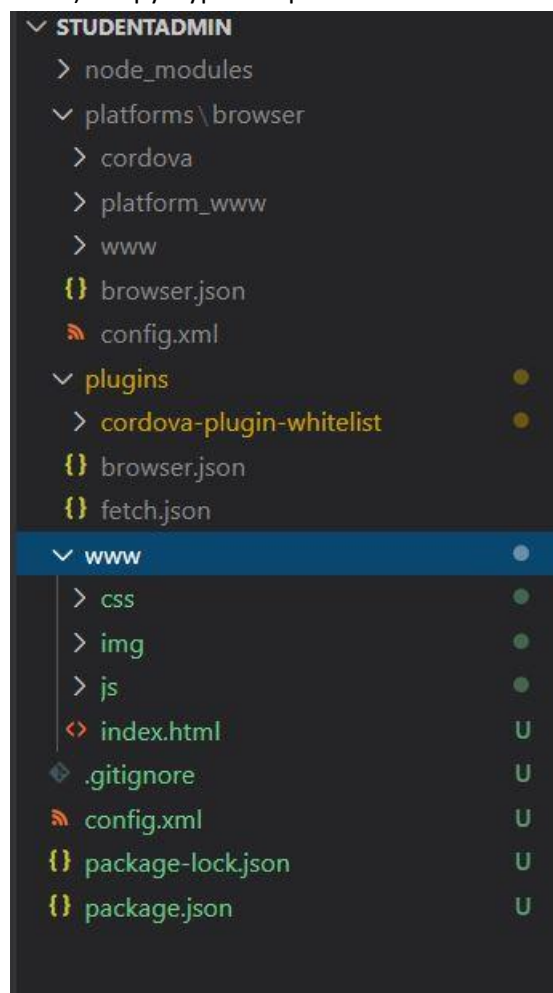
приложението трябва да изглежда така:



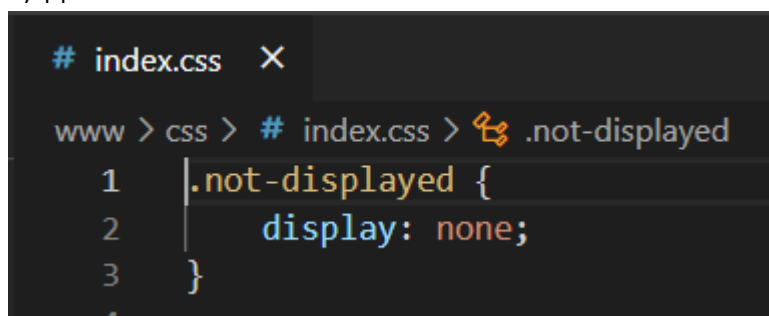
4. The Student Admin App

Следната задача, ще обясни как стъпка-по-стъпка да създадем хибридно мобилно приложение, което ще управлява профилите ни в [портала](#). Приложението ще изглежда така:

1) Структура на проекта



2) Добавяме в index.css



3) Промени в **index.html** следната стойност:

```
<meta http-equiv="Content-Security-Policy" content="default-src *; style-src 'self' http://* 'unsafe-inline'; img-src 'self' data:; script-src 'self' http://* 'unsafe-inline' 'unsafe-eval'" />
```

4) Премахни

```
<!-- page 2 -->
<div data-role="page" id="page2">
  <div data-role="header">
    <a href="#" data-icon="back" data-rel="back" title="Go back">Back</a>
    <h1>Page2</h1>
  </div>

  <div class="ui-content">
    <p>This is page 2</p>
  </div>
  <!-- end page 2 content -->

  <div data-role="footer" data-position="fixed">
    <h1>Page2</h1>
  </div>
  <!-- /footer -->
</div>
<!-- /page2 -->

<!-- page 3 -->
<div data-role="page" id="page3">
  <div data-role="header">
    <a href="#" data-icon="back" data-rel="back" title="Go back">Back</a>
    <h1>Page3</h1>
  </div>

  <div class="ui-content">
    <p>This is page 3</p>
  </div>
  <!-- end page 3 content -->

  <div data-role="footer" data-position="fixed">
    <h1>Page3</h1>
  </div>
  <!-- /footer -->
</div>
<!-- /page3 -->
```

5) Промени **homePage** по следния начин:

```
<body>
  <!-- home page -->
  <div data-role="page" id="homePage">
    <div data-role="header">
      <h1>Student Admin</h1>
    </div>
    <ul data-role="listview">
      <li><a href="#detailsPage">Student Details</a></li>
    </ul>

    <div data-role="footer" data-position="fixed">
      <h1>Home Page</h1>
    </div>
    <!-- /footer -->
  </div>
  <!-- home page -->
```

6) **page 1** ще бъде преобразен напълно от

```
<!-- page 1 -->
<div data-role="page" id="detailsPage">
  <div data-role="header">
    <a href="#" data-icon="back" data-rel="back" title="Go back">Back</a>
    <h1>Page1</h1>
  </div>

  <div class="ui-content">
    <p>Page 1 content goes here</p>
  </div>
  <!-- end page 1 content -->

  <div data-role="footer" data-position="fixed">
    <h1>Page1</h1>
  </div>
  <!-- /footer -->
</div>
<!-- /page1 -->
```

```
<!-- Details Page -->
<div data-role="page" id="detailsPage">
  <div data-role="header">
    <a href="#" data-icon="back" data-rel="back" title="Go back">Back</a>
    <h1 id="student-name">Student Name</h1>
  </div>

  <div class="ui-content">
    <img src="" alt="student-picture-view" id="student-picture-view" height="200" width="200">
    <br>

    <label for="student-faculty-number">Faculty Number:</label>
    <p id="student-faculty-number">Student Faculty Number</p>

    <label for="student-email">Email:</label>
    <p id="student-email">Student Email</p>

    <label for="student-username">User Name:</label>
    <p id="student-username">Generated User Name</p>
  </div>

  <div data-role="footer" data-position="fixed">
    <h4>
      Make picture
      <a href="#picturePage" data-transition="slide" data-role="button" data-icon="camera">Camera</a>
    </h4>
  </div>
</div>
<!-- /Details -->
```

7) Добавете също и

```
<!-- Picture Page -->
<div data-role="page" id="picturePage">
  <div data-role="header">
    <a href="#" data-icon="back" data-rel="back" title="Go back">Back</a>
    <h1>Student Picture</h1>
  </div>

  <div class="ui-content">
    <div>
      <button id="camera-btn" data-role="button" data-icon="camera">CLICK</button>
      

      <label for="facultyNumber">Faculty Number:</label>
      <input type="text" id="faculty-number" min="1" />

      <label for="password">Password:</label>
      <input type="password" id="password" />

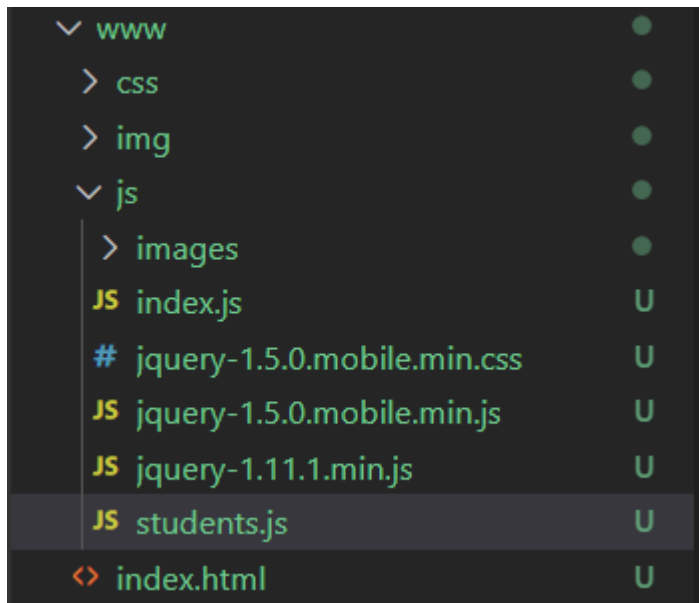
      <button id="addPicture">Add Picture</button>
    </div>
  </div>

  <div data-role="footer" data-position="fixed">
    <h1>Student Picture</h1>
  </div>
</div>
<!-- /Picture -->
```

8) В папка js създайте **students.js** файл, който да има референция в **index.html** .

```
<script type="text/javascript" src="js/jquery-1.5.0.mobile.min.js"></script>
<script src="js/index.js"></script>
<script src="js/students.js"></script>
</body>
```

9) Трябва да изглежда така. Images папката може да ги вземете от [ТУК](#).



10) Добавяне на глобални константни стойности:

```
const baseSurviceUrl = "https://students-manager.azurewebsites.net/api/students/";
const username = "guest";
const password = "guest";
const base64Auth = btoa(username + ":" + password);
```

11) Имплементиране на метода, който да зарежда стойностите на началния екран:

```
function loadStudents() {
    $.mobile.loading("show");

    $.ajax({
        type: "GET",
        url: baseSurviceUrl,
        beforeSend: function(xhr) {
            xhr.setRequestHeader("Authorization", "Basic " + base64Auth);
        },
        success: function(students) {
            $.mobile.loading("hide");
            displayStudents(students);
        },
        error: function(XMLHttpRequest, textStatus, errorThrown) {
            alert("Status: " + textStatus);
            alert("Error: " + errorThrown);
        }
    });
}
```

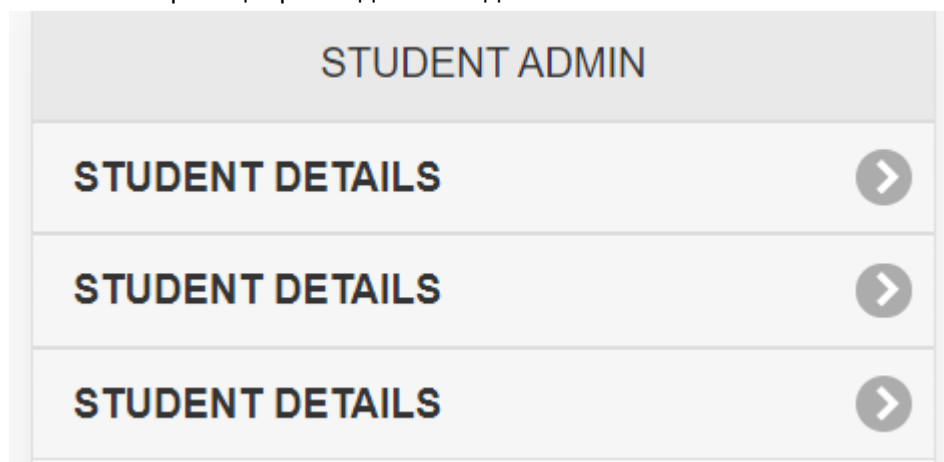

12) Callback функция:

```
function displayStudents(students) {  
    $('#results').empty();  
  
    for (let student of students) {  
        $('#results').append(  
            $('<li>').append(  
                $('<a>')  
                .attr('href', '#detailsPage')  
                .addClass("ui-btn ui-btn-icon-right ui-icon-carat-r")  
                .click(function() { setStudentDetails(student); })  
                .append(student.fullName));  
            }  
        }  
    }
```

13) Променяме **index.js** да изглежда по следния начин:

```
document.addEventListener('deviceready', onDeviceReady, false);  
  
function onDeviceReady() {  
    console.log('Running cordova-' + cordova.platformId + '@' + cordova.version);  
    // Home & Details Pages  
    loadStudents();  
  
    // Picture  
    $('#camera-btn').click(getPicture);  
    $('#addPicture').click(sendPictureRequest);  
}
```

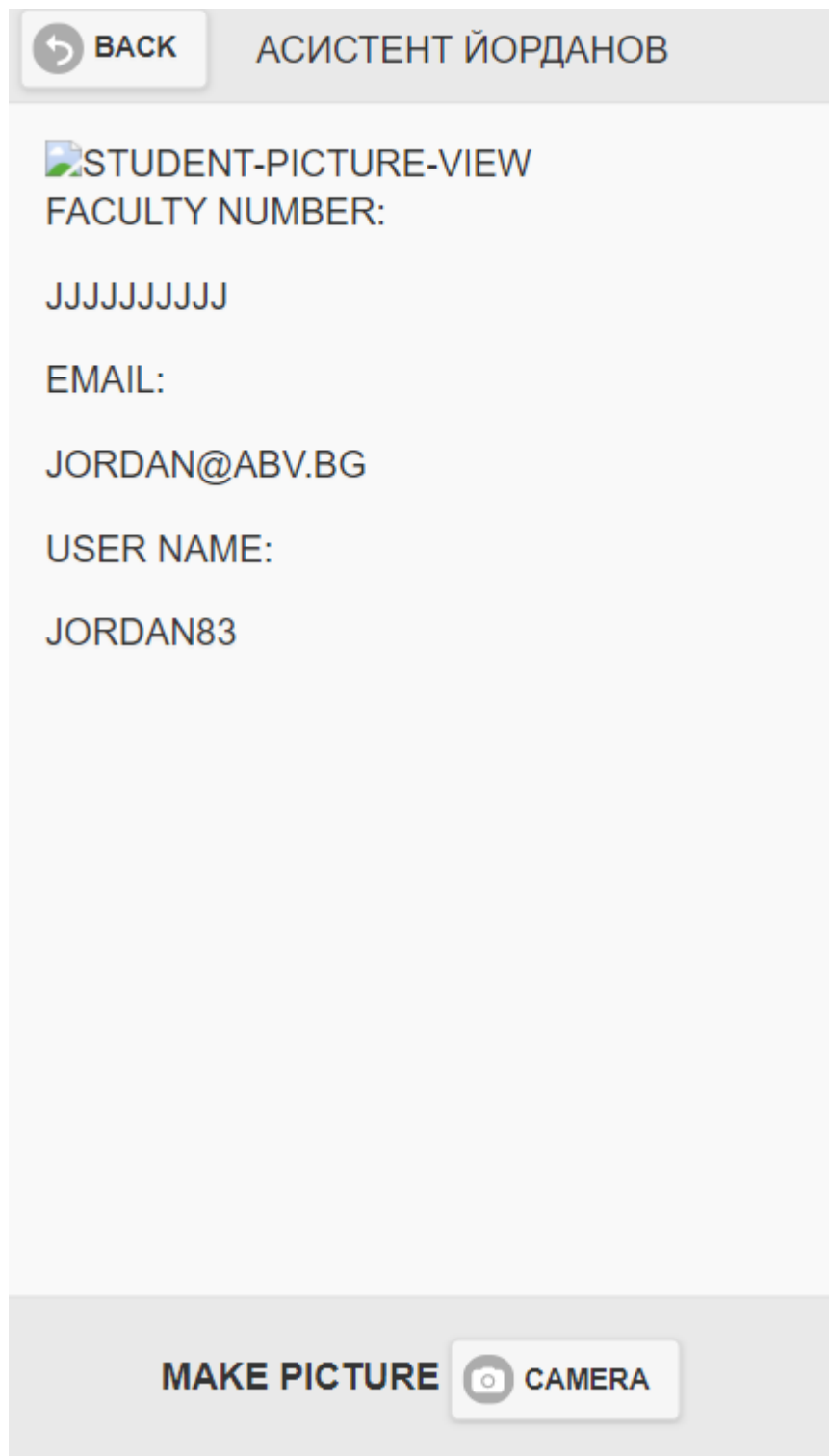
14) Началната страница трябва да изглежда така:



15) Имплементирайте следния метод, който да задава стойностите в **details page**

```
function setStudentDetails(student) {  
    $("#student-name").text(student.fullName);  
    $("#student-picture-view").attr("src", student.base64EncodePicture);  
  
    $('#student-faculty-number').text(student.facultyNumber);  
    $('#student-email').text(student.email);  
    $('#student-username').text(student.userName);  
}
```

16) Трябва да изглежда така:



The screenshot shows a web application interface. At the top, there is a header bar with a "BACK" button (represented by a left arrow icon) and the text "АСИСТЕНТ ЙОРДАНОВ". Below the header, the main content area displays the following information: "STUDENT-PICTURE-VIEW" (with a small image icon to the left), "FACULTY NUMBER:", "JJJJJJJJJJ", "EMAIL:", "JORDAN@ABV.BG", "USER NAME:", and "JORDAN83". At the bottom of the page, there is a footer bar with the text "MAKE PICTURE" and a "CAMERA" button (represented by a camera icon).

17) **picturePage** страницата трябва да изглежда така:

BACK STUDENT PICTURE

CLICK

FACULTY NUMBER:

PASSWORD:

Add Picture

STUDENT PICTURE

18) Изпълнете командата: `cordova plugin add cordova-plugin-camera`

```
PS C:\Users\yoyordan\Documents\GitHub\hybrid-app-development\04. Cordova jQuery\StudentAdmin> cordova plugin add cordova-plugin-camera
6.0.0
Installing "cordova-plugin-camera" for browser
6.0.0
Adding cordova-plugin-camera to package.json
```

19) Функциите стоящи зад тази страница са:

```
function getPicture() {
    navigator.camera.getPicture(succeededCameraCallback, failedCameraCallback, {
        quality: 25,
        destinationType: Camera.DestinationType.DATA_URL
    });
}

function succeededCameraCallback(imageData) {
    $('#myImage').attr('src', 'data:image/jpeg;base64,' + imageData);
    $('#myImage').css('display', 'block');
    $('#myImage').show();
}

function failedCameraCallback(message) {
    alert(message);
}

function sendPictureRequest() {
    $.mobile.loading("show");

    let base64Img = $('#myImage').attr('src');
    let facultyNumber = $('#faculty-number').val();
    let password = $('#password').val();

    let requestData = {
        FacultyNumber: facultyNumber,
        Password: password,
        Picture: base64Img
    };
    console.log(requestData);

    $.ajax({
        type: "PUT",
        url: baseServiceUrl + "picture",
        contentType: "application/json",
        dataType: "json",
        data: JSON.stringify(requestData),
        success: function() {
            $.mobile.loading("hide");
            alert("Successfully added!");
            loadDetailsPage();
        },
        error: function(XMLHttpRequest, textStatus, errorThrown) {
            $.mobile.loading("hide");
            loadDetailsPage();
        }
    });
}
```

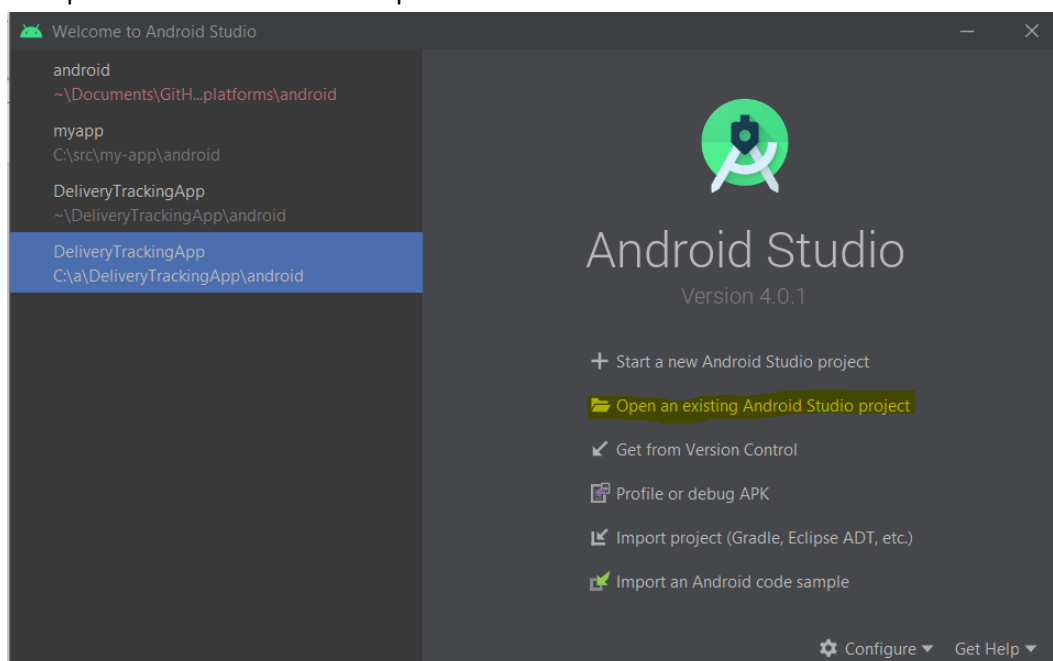
5. Student Admin за Android

Ако имате телефон, който работи с операционна система Android, моля, изтеглете си [Android Studio](#), и прочетете [Ръководство за платформа за Android на Apache Cordova](#).

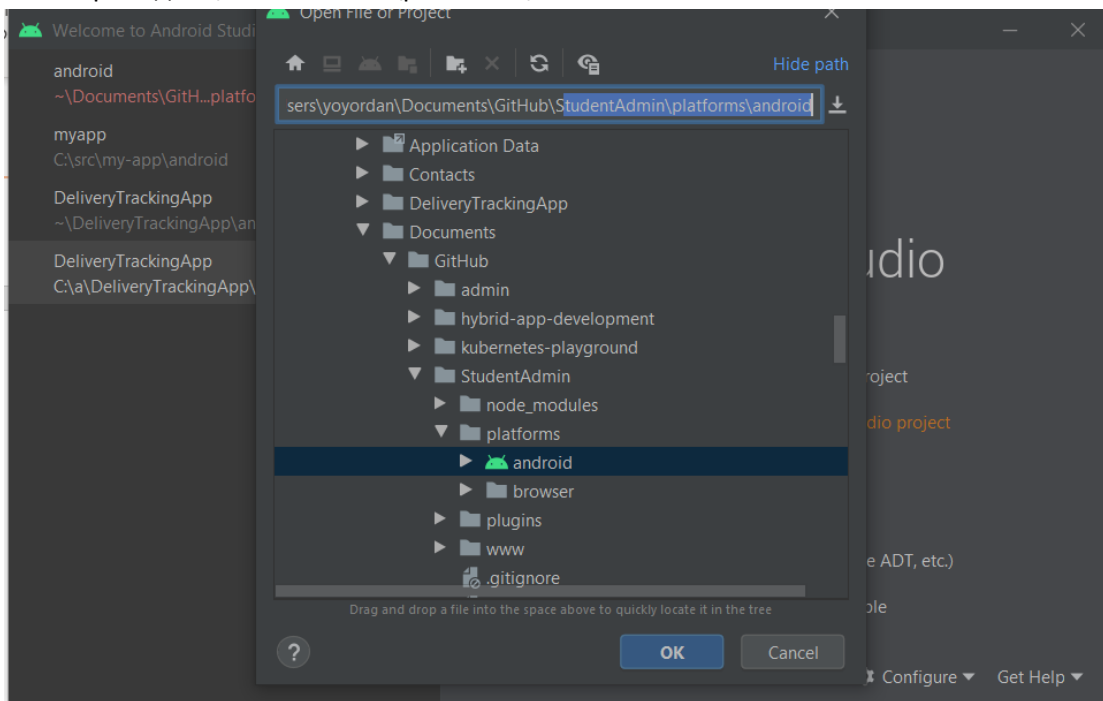
- 1) След като настроите всички глобални променливи и имате инсталирана Java ,
В директорията на StudentAdmin изпълнете командата *cordova platform add android*.

```
PS C:\Users\yoyordan\Documents\GitHub\StudentAdmin> cordova platform add android
Using cordova-fetch for cordova-android@9.0.0
Adding android project...
Creating Cordova project for the Android platform:
  Path: platforms\android
  Package: io.cordova.hellocordova
  Name: HelloCordova
  Activity: MainActivity
  Android target: android-29
Subproject Path: CordovaLib
Subproject Path: app
Android project created with cordova-android@9.0.0
Installing "cordova-plugin-camera" for android
Subproject Path: CordovaLib
Subproject Path: app
Installing "cordova-plugin-whitelist" for android
PS C:\Users\yoyordan\Documents\GitHub\StudentAdmin>
```

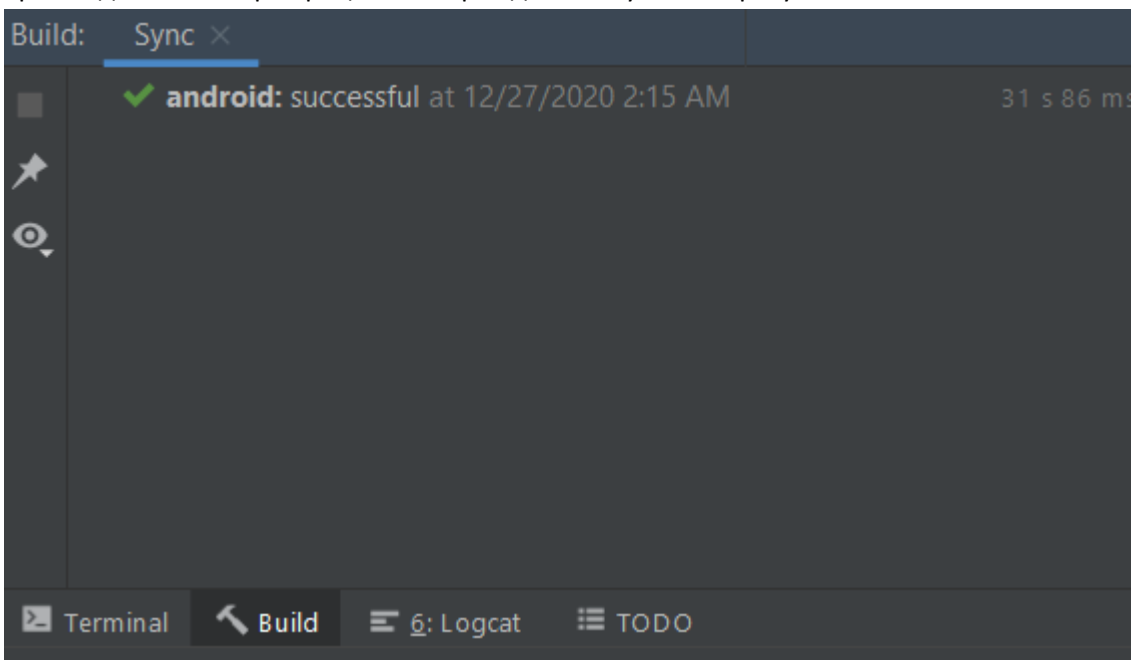
- 2) Отваряте Android Studio и избирате :



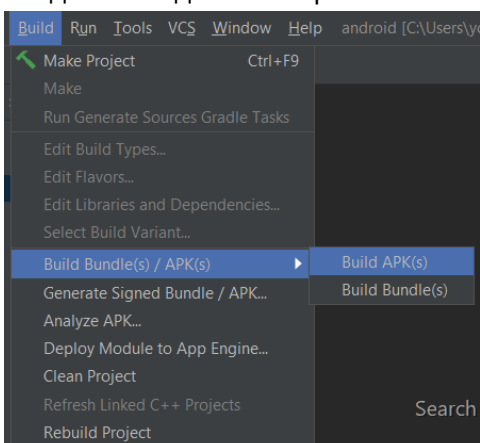
3) Навигирате до `..\StudentAdmin\platforms\android`



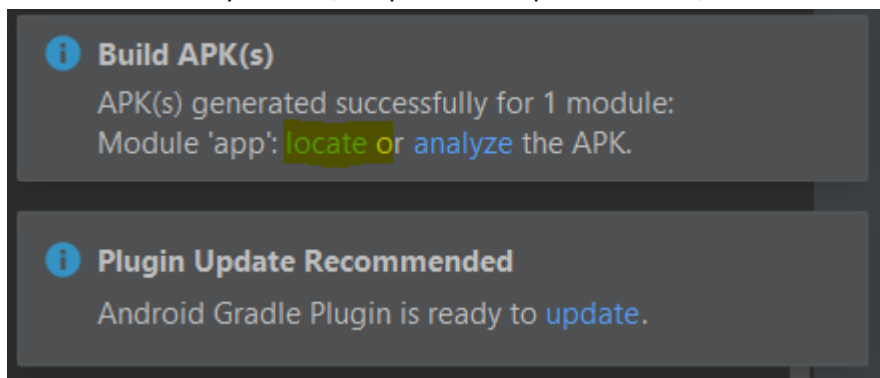
4) Трябва да се активира процес по изграждане със успешен резултат:



5) След това създайте APK файл:



б) Ако всичко мине успешно, свържете телефона си с USB, към компютъра. Навигирайте до



И копирайте app-debug.apk в Internal shared storage на телефона си.
Последните 3 стъпки се осъществяват чрез мобилното устройство:

