**Part1-Project Design & Description Document**

1. Description of the project (5 marks)

On the one of November 2018, we were request to design and develop a cloud database for the Latifa Hospital, which includes database and several collections. This cloud database will help doctors and Patients to take an appointment online and check the schedule of doctor and which them is available. System will have updated the all-new dates.

The purpose for this system is to improve the hospital appointments by using cloud app includes new features that help doctors and patient. For the patient it will be easy to use and take online appointments. However, it helpful for reception because it saves their time by just conform their appointments.

This application allow doctors to see all the transactions by using this app. The benefits of the new system is saving time, faster, store information by using this app.

2. Design of cloud database – names of collections and hosting details (2.5 marks)

|  |  |
| --- | --- |
| Database | * Appointment |
| Collections | * Patients * Doctors * Appointment |
| Patients | * Patient ID * First Name * Last Name * Phone Number * Data of birth * Doctor ID |
| Doctors | * Doctor ID * First Name * Last Name * Phone Number * Salary * Address * Specialization |
| Appointment | * Doctor ID * Patient name * Date & Time * Gender * Country |

3. Design of the routes and the sample data returned by them (5 marks)

|  |  |
| --- | --- |
| Patients | * Patient ID: “123”   First Name: “Sara”  Last Name: “AlShamisi”  Phone Number: “050-1212123”  Date of birth: “10-Oct-1997”  Doctor ID: “101”   * Patient ID: “456”   First Name: “Mariam”  Last Name: “Alhosani”  Phone Number: “055-3434345”  Date of birth: “07-Jan-1990”  Doctor ID: “102”   * Patient ID: “789”   First Name: “Sultan”  Last Name: “Alhammadi”  Phone Number: “055-67667678”  Date of birth: “20-Oct-1989”  Doctor ID: “103”   * Patient ID: “110”   First Name: “Talal”  Last Name: “Alraeesi”  Phone Number: “055-5050505”  Data of birth: “30-Jun-1997”  Doctor ID:”104”   * Patient ID: “220”   First Name: “Aisha”  Last Name: “Alsuwaidi”  Phone Number: “050-8887776”  Data of birth: “06-Fab-1986”  Doctor ID: “105” |
| Doctors | * Doctor ID: “101:   First Name: "Mohammed"  Last Name:  "Alshamsi"  Phone Number: "050-1233456"  Salary: 25,000  Address: "Sharjah-AlNouf"  Specialization: " Dentist"   * Doctor ID: “102”   First Name: "Hana"  Last Name:  "Almessabi"  Phone Number: "050-6578378"  Salary: "20,000"  Address: "Sharjah-AlJuraina"  Specialization:  "Laboratory Medicine"   * Doctor ID: “103”   First Name: "Misone"  Last Name: "Abushaala"  Phone Number:  "050-3456543"  Salary: "20,000"  Address: "Sharjah"  Specialization: “Senior Specialist Ophthalmologist"   * Doctor ID: ”104”   First Name: "Tahani"  Last Name: "Atif"  Phone Number: "050-6485637"  Salary: "20,000"  Address: "Sharjah-AlNhdaa"  Specialization: "Specialist Family Medicine"   * Doctor ID: “105”   First Name: "Mohammed"  Last Name: "Murad"  Phone Number:  "050-4553345"  Salary: "20,000"  Address:  "Sharjah-AlNouf"  Specialization: "Consultant Oral Medicine" |
| Appointment | * Doctor ID: “105”   Patient name: “Aisha Alsuwaidi”  Date & Time: “20-Nov-2018”  Gender: “Female”  Country: “UAE”   * Doctor ID: ”104”   Patient name: “Talal Alraeesi”  Date & Time: “15-Dec-2018”  Gender: “Male”  Country: “UAE”   * Doctor ID: “101”   Patient name: “Sara AlShamisi”  Date & Time: “12-Nov-2018”  Gender: “Female”  Country: “UAE”   * Doctor ID: “102”   Patient name: “Mariam Alhosani”  Date & Time: “05-Dec-2018”  Gender: “Female”  Country: “UAE”   * Doctor ID: “101”   Patient name: “Abdullah Alhammadi”  Date & Time: “25-Dec-2018”  Gender: “Male”  Country: “UAE” |

4. Architecture diagram with brief explanation (2.5 marks)

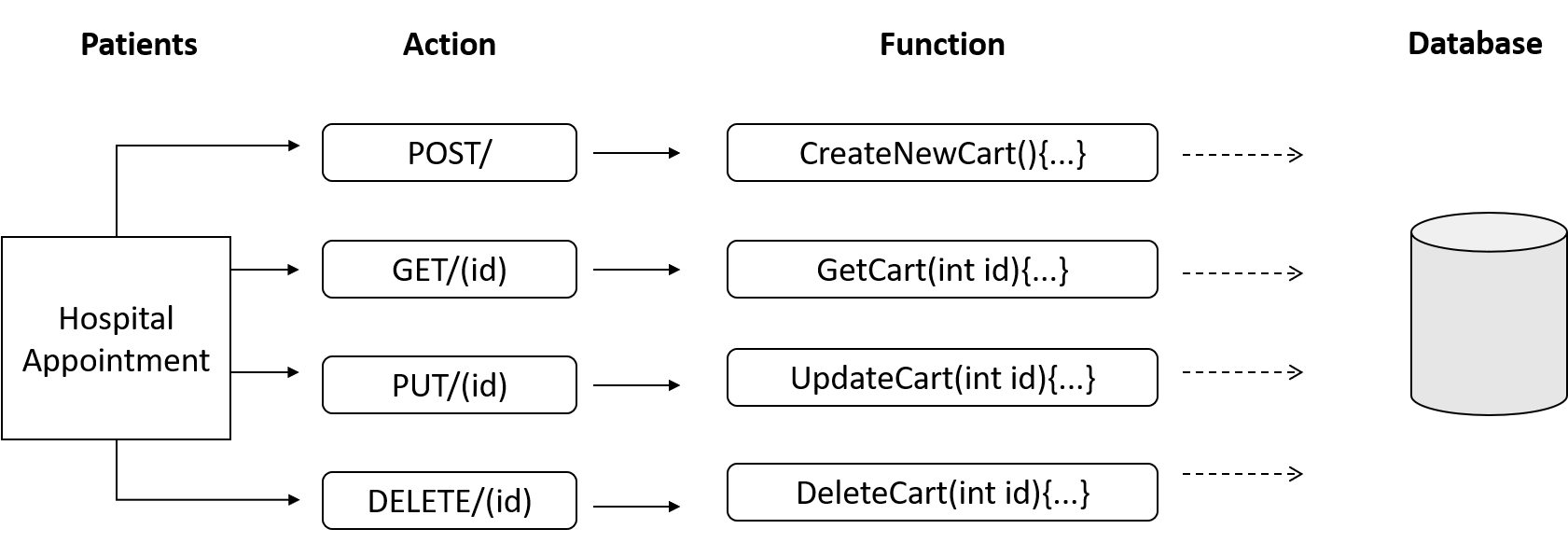


Figure1: Application and cloud database architecture based on REST

The Application and cloud database architecture based has divided into three tiers, which are:

* Cloud database tier
* Application tier
* Client tier

Definition each tier:

|  |  |
| --- | --- |
| Cloud database tier | Database servers store and retrieve the information. Data in this tier is kept independent of application servers or business logic. |
| Application tier (web) | This tier also called the middle tier, logic tier, business logic or logic tier, it is drawn from the presentation tier and it controls application operations by performing detailed processing. This tier run in several programs such as node.js, IIS, Apache, Java EE, ASP.NET, etc. |
| Client tier (web UI or mobile app) | Occupies the top level. This tier communicates with other tiers by sending results to the browser and show information related to services available or ready on a website. it's run in IE, Chrome, Firefox, etc. |

Websites and mobile applications are clients, which invoke the REST API through the HTPP methods, GET, POST, PUT and DELETE, the code for this function in your express.js then calls database functions to retrieve, insert, update or delete the data from the cloud database.

**Part 2 - Implementation**

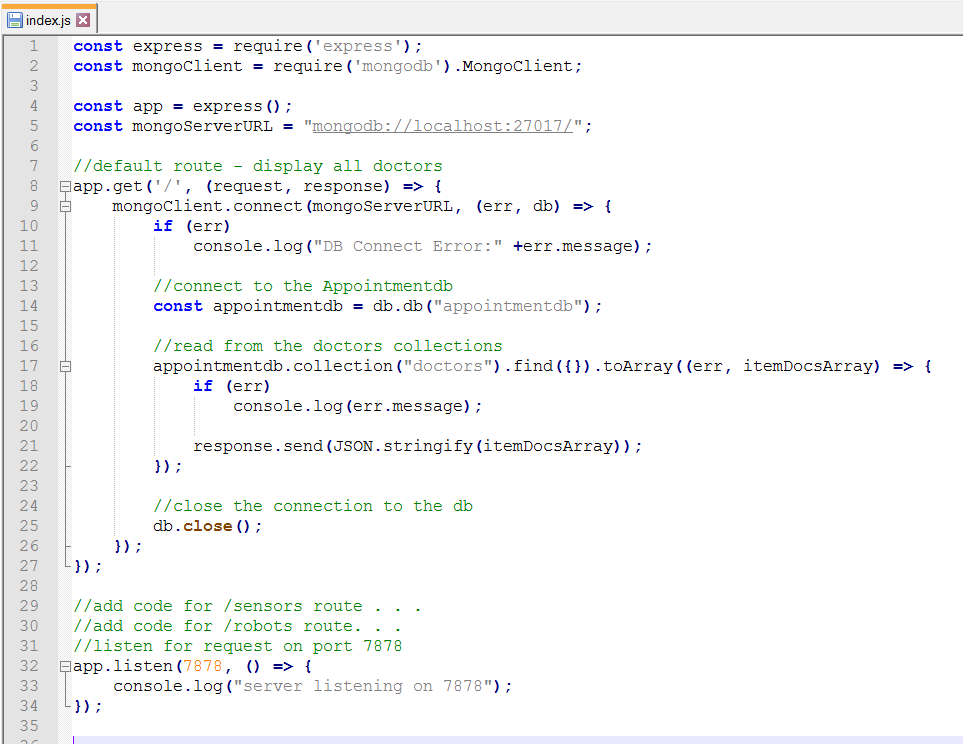
1.Use a cloud database to store data (such as mLab, Firebase etc.) [ 10 marks]

Store Data in MongoDB

|  |
| --- |
|  |

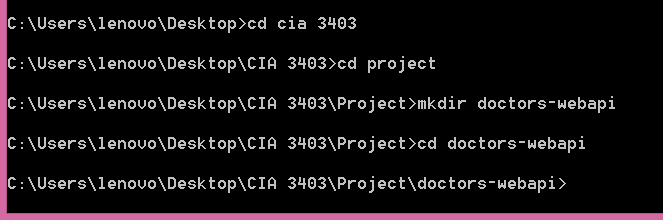
2. Use node.js and express.js to build at least five routes using your cloud database including one for adding data to the cloud data and one for updating data on the cloud data [20 marks]

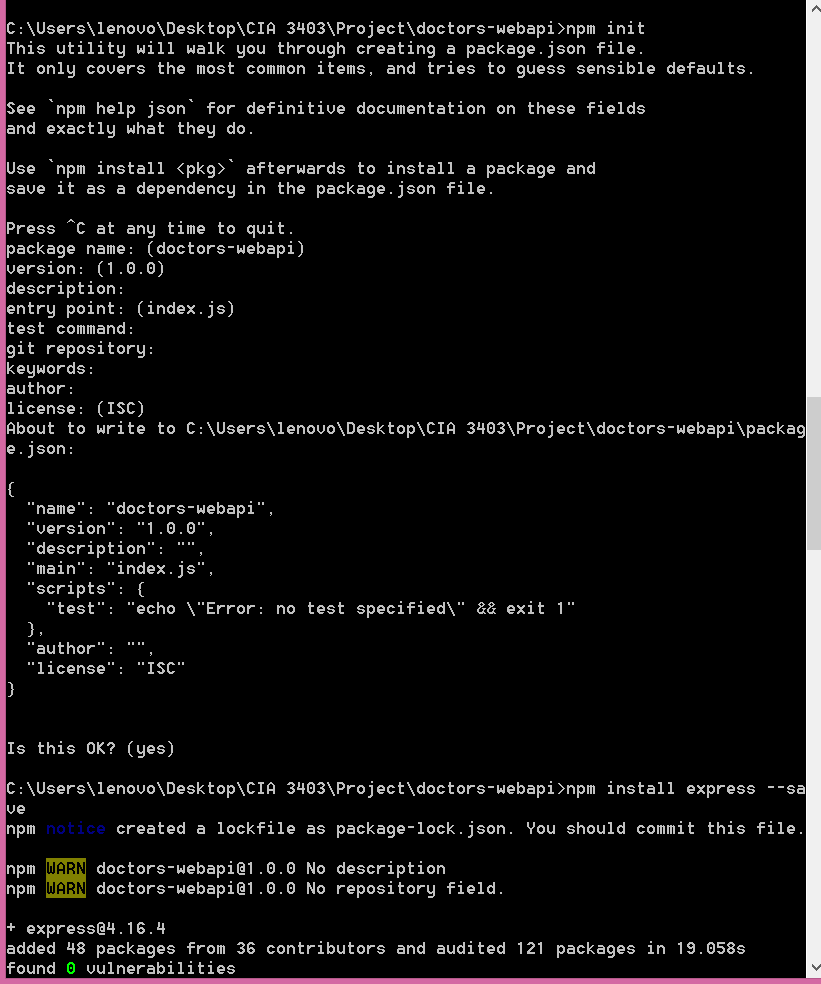
Node.js”index.js”:

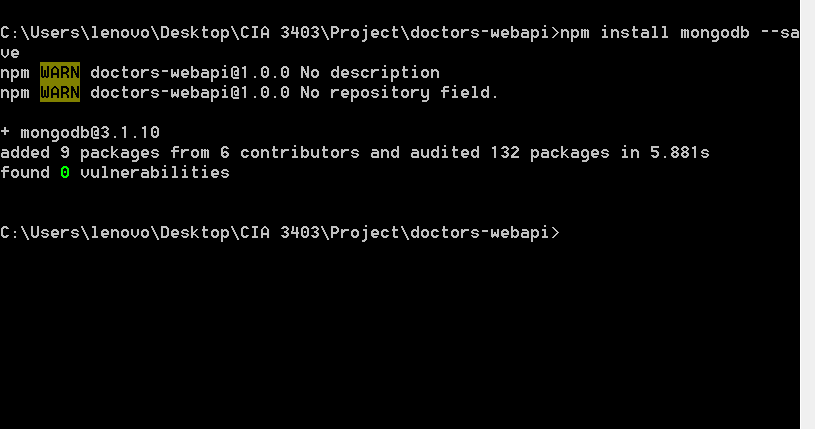


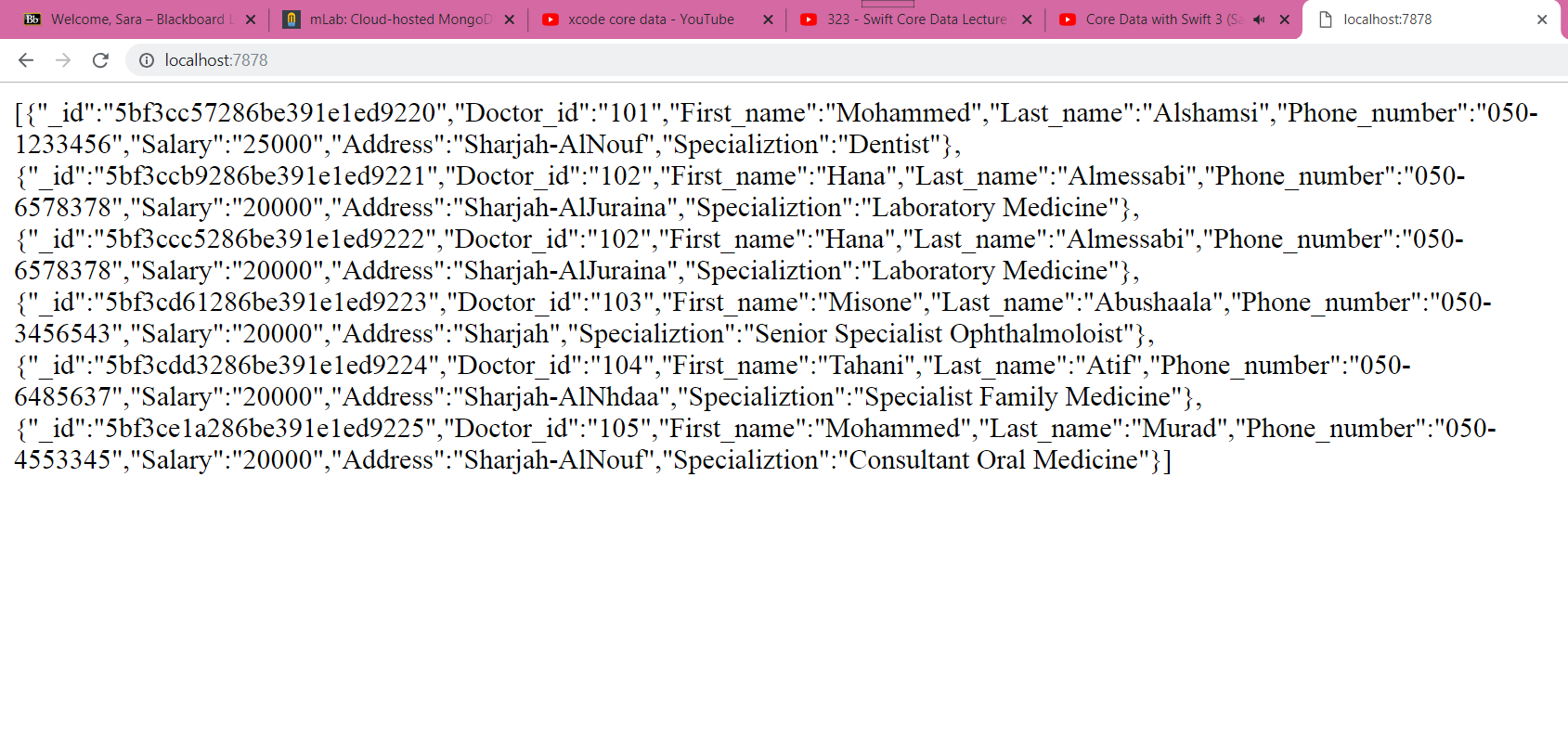
1. Route

* Make directory for doctors



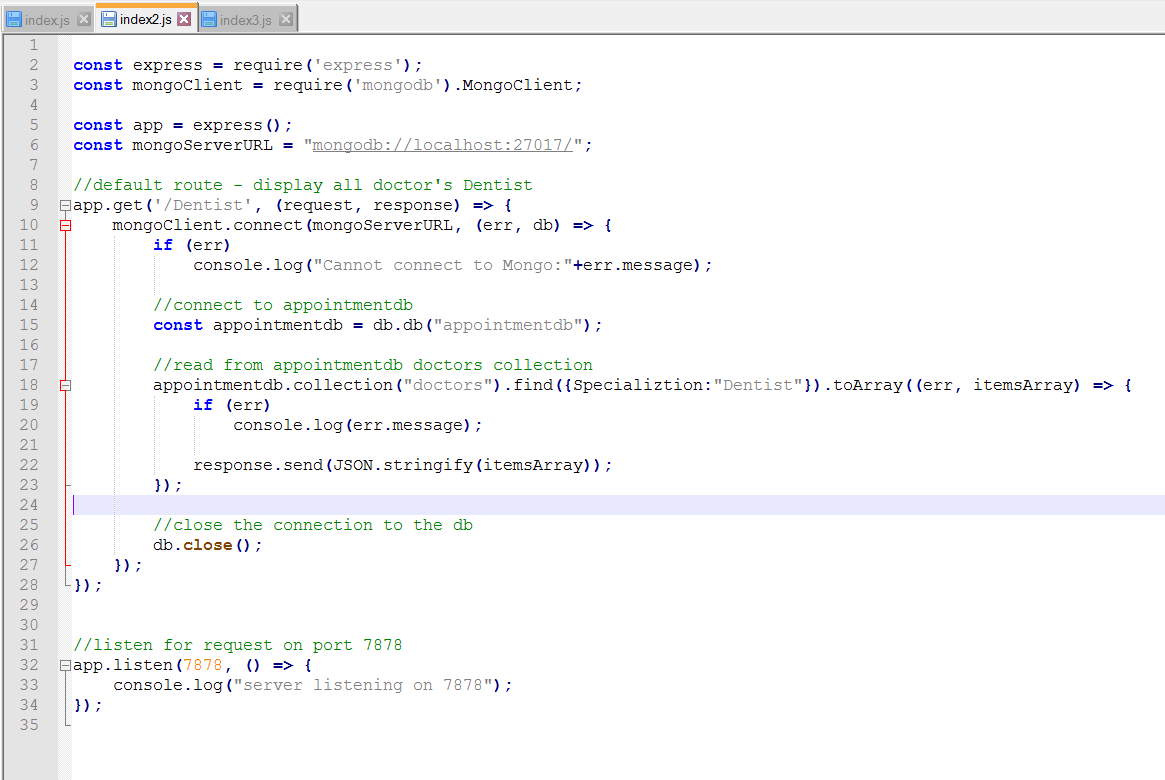


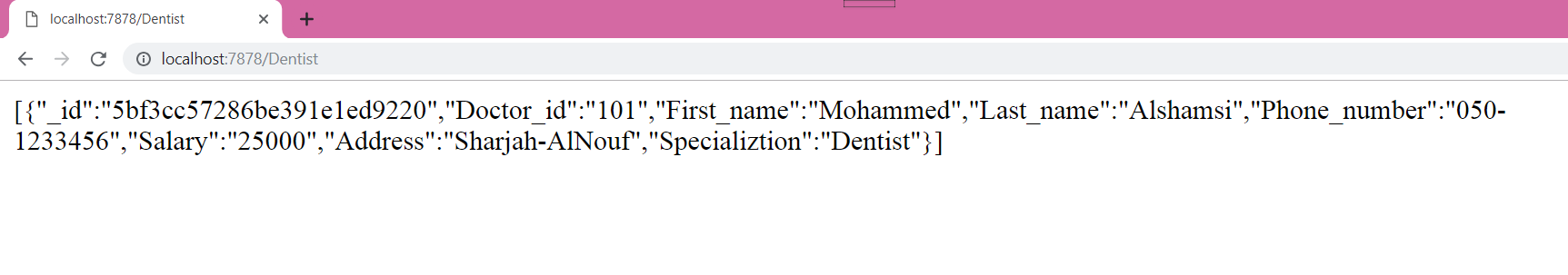




1. Route

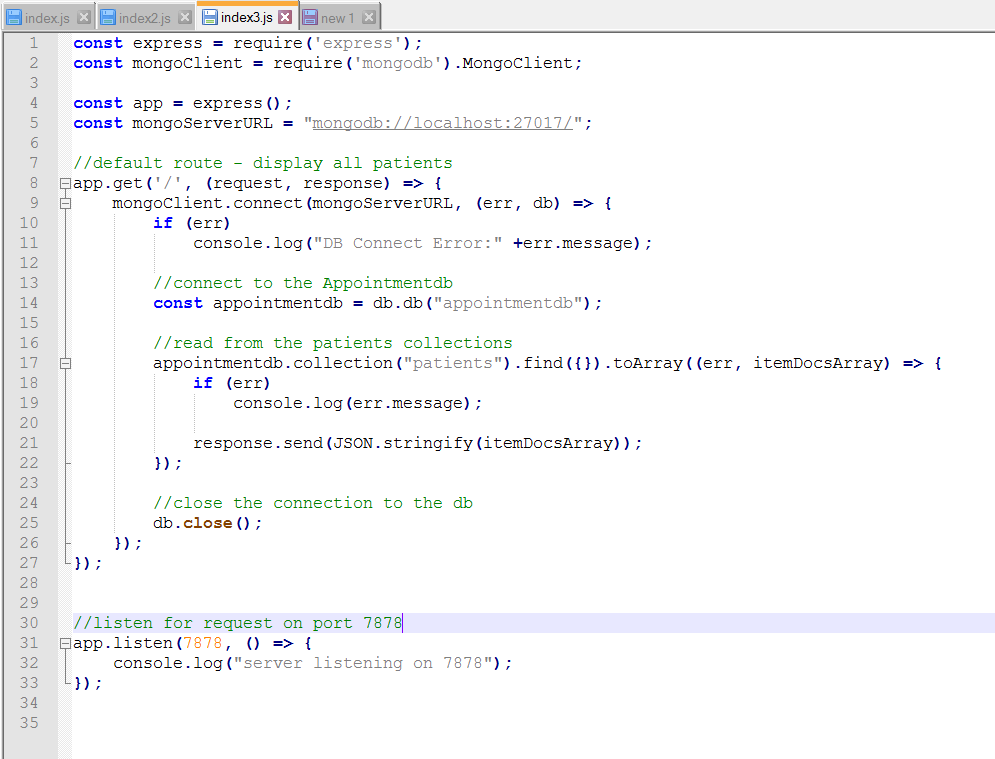
Node.js”index2.js”:

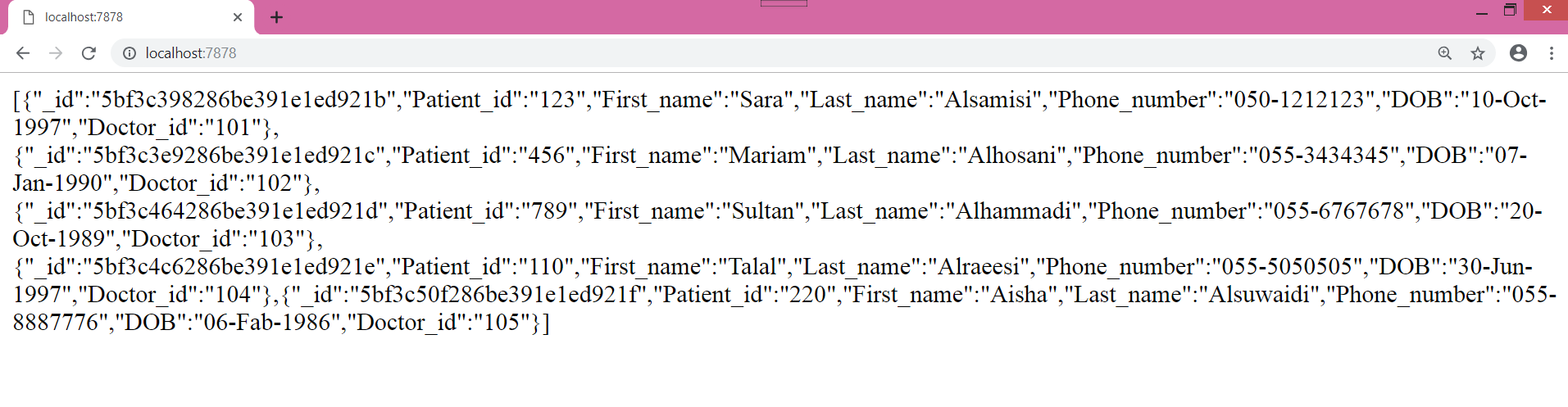




1. Route

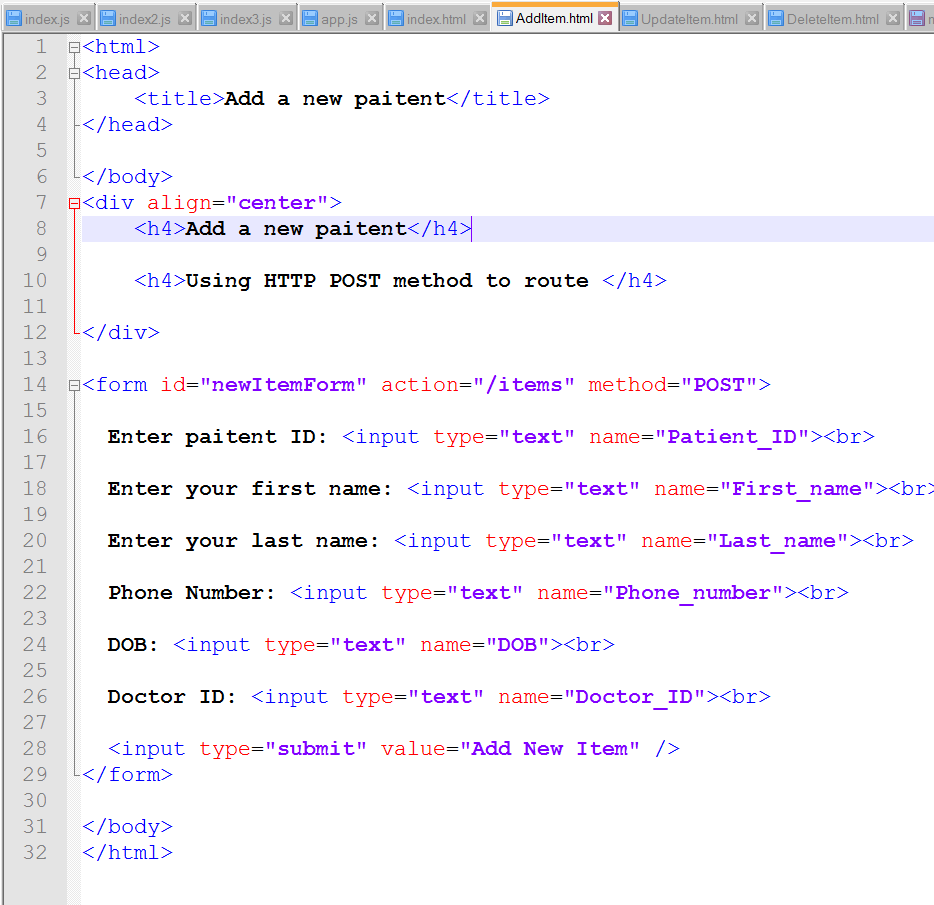
Node.js”index3.js”:

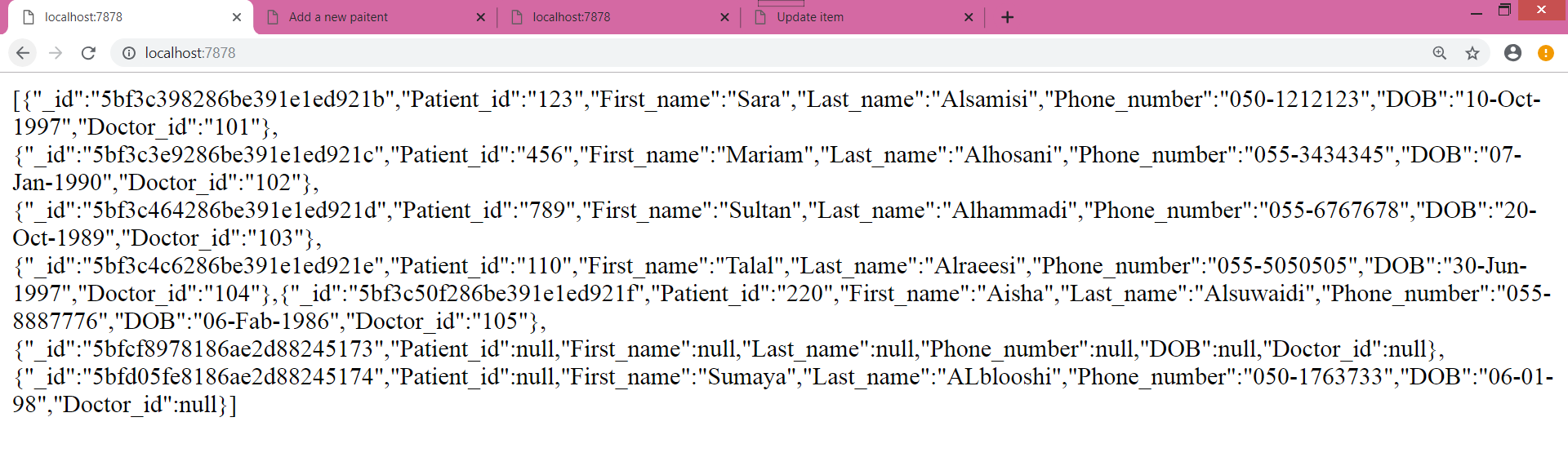


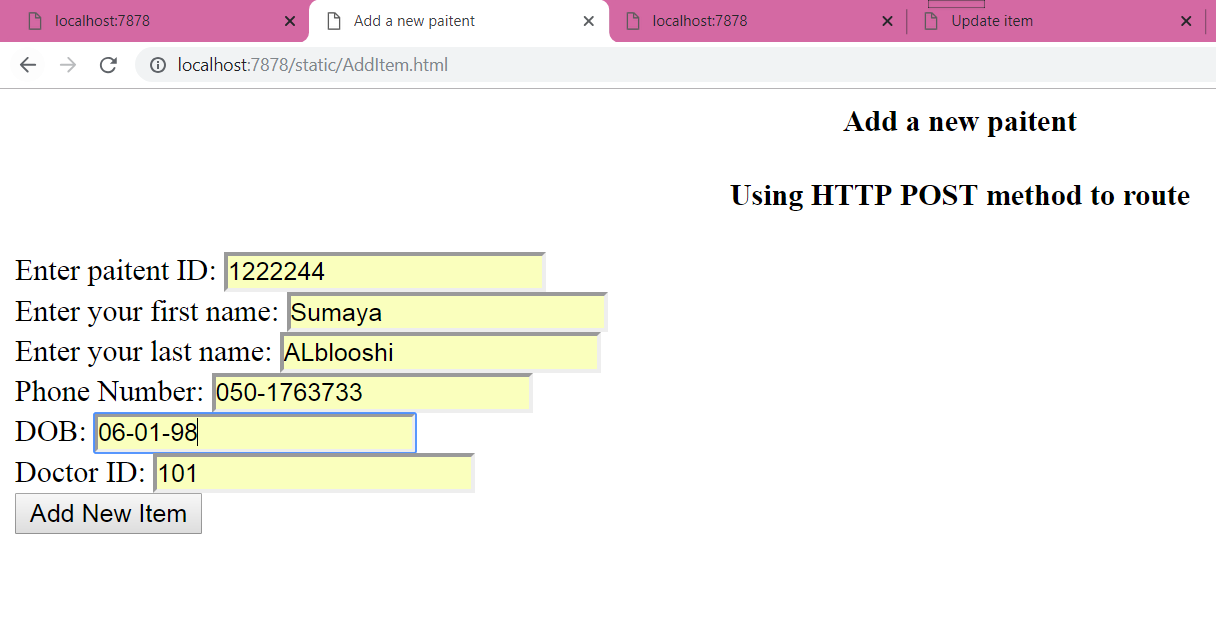


1. Route

Node.js”index.js”:



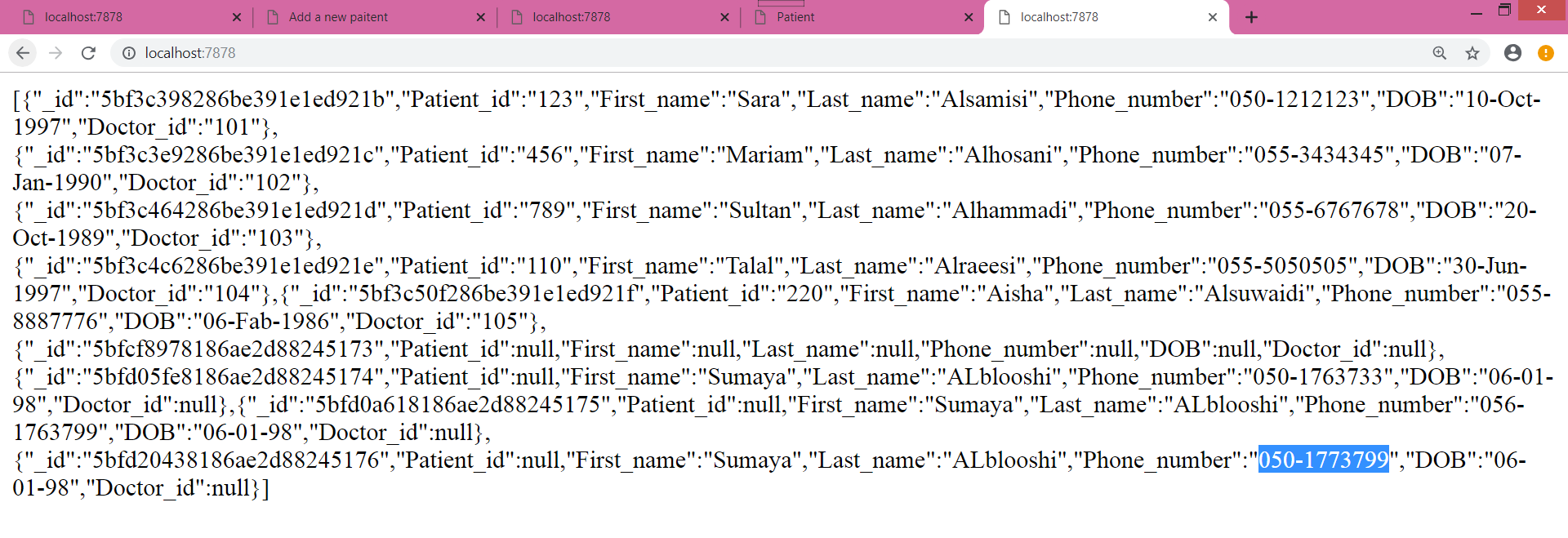




1. Route

Node.js”index.js”:

|  |
| --- |
| <html>  <head>  <title>Update patient information</title>  </head>  </body>  <div align="center">  <h4>Update </h4>  <h4>Using HTTP PUT to route <code>/items</code></h4>  </div>  <div>Note: that patient name must exist</div>  <form id="updateItemForm" action="/patients" method="POST">    Enter paitent ID: <input type="text" name="Patient\_ID"><br>    Enter your first name: <input type="text" name="First\_name"><br>    Enter your last name: <input type="text" name="Last\_name"><br>    Phone Number: <input type="text" name="Phone\_number"><br>    DOB: <input type="text" name="DOB"><br>    Doctor ID: <input type="text" name="Doctor\_ID"><br>    <input type="button" value="Load information" onclick="loadData()"/>  &nbsp;&nbsp;&nbsp;  <input type="submit" value="Update info" />  </form>  <script>  function handleResponse() {  console.log(this.responseText);  let updateForm = document.getElementById("updateItemForm");  //convert responseText to a JS object  let jsonArray = JSON.parse(this.responseText);  if (jsonArray.length == 0) {  alert("No data found for patient " + updateForm.elements["First\_name"].value);  return;  }    //get the references to the textboxes and display  //the data from server inside them.  let tNum = updateForm.elements["Phone\_number"];  tNum.value = jsonArray[0].Phone\_number;  }  //load the item data from server  function loadData() {  let updateForm = document.getElementById("updateItemForm");  let FName = updateForm.elements["First\_name"].value;  let httpRequest = new XMLHttpRequest();  httpRequest.addEventListener("load", handleResponse);  httpRequest.open("GET", "http://localhost:7979/items/"+itemName);  httpRequest.send();  }  </script>  </body>  </html> |
|  |

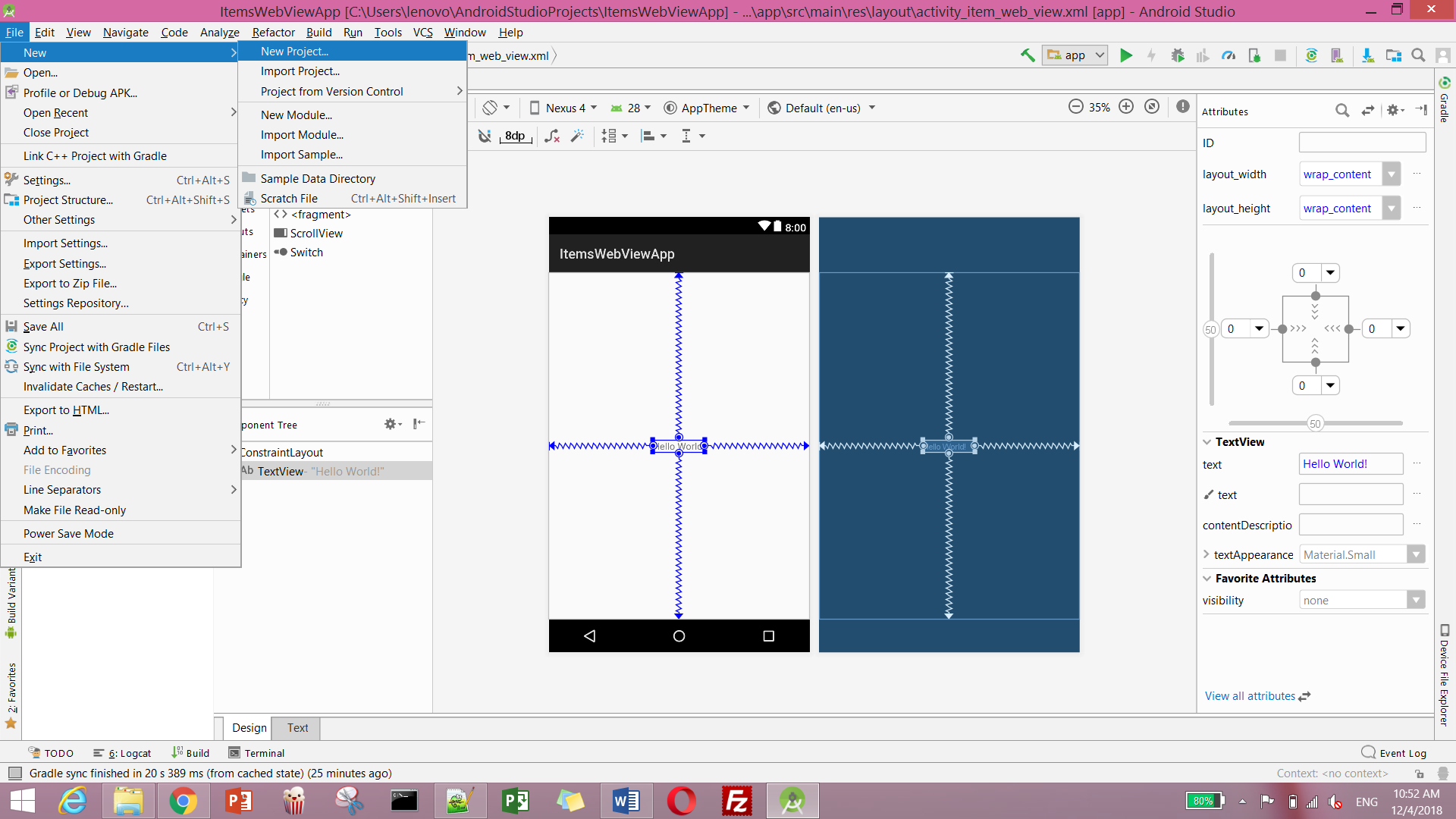


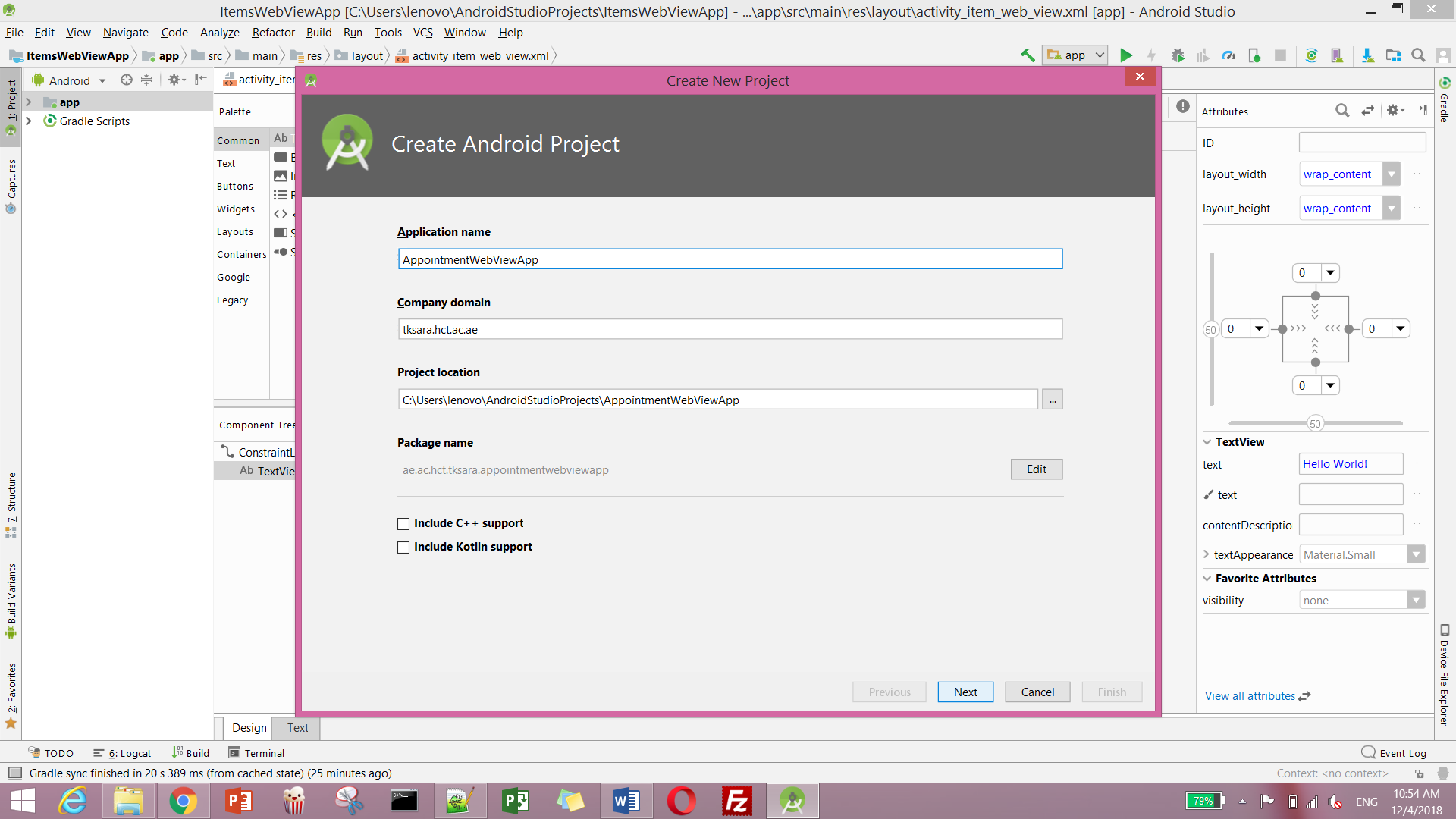
3. Build web pages to display the data using the routes – use XHR [10 marks]

|  |
| --- |
|  |
|  |
|  |

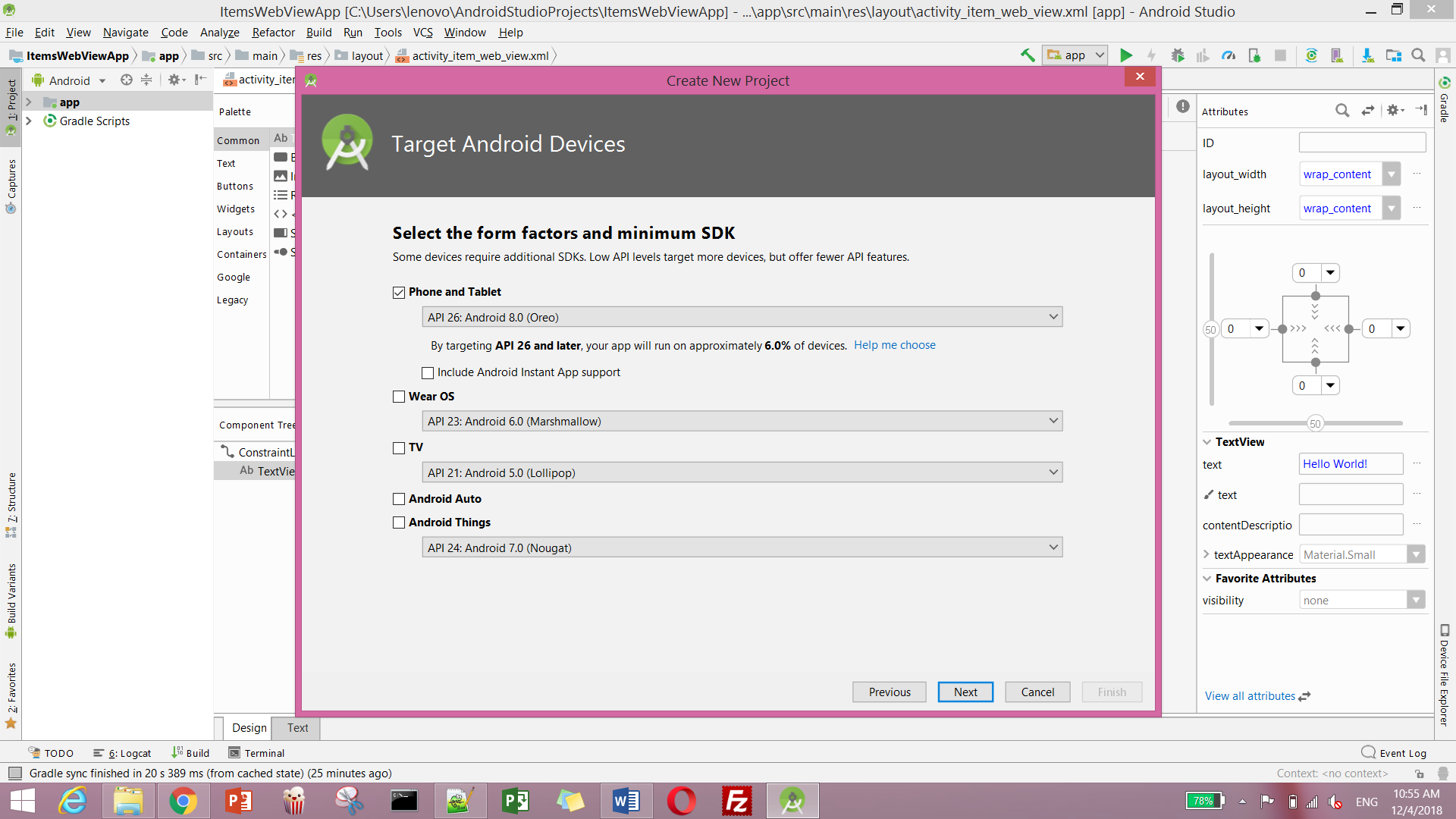
4. Build a mobile app to show the data in cloud data using different REST API routes, add one screen to add the data and one screen to update data [20 marks]

Create a new android project

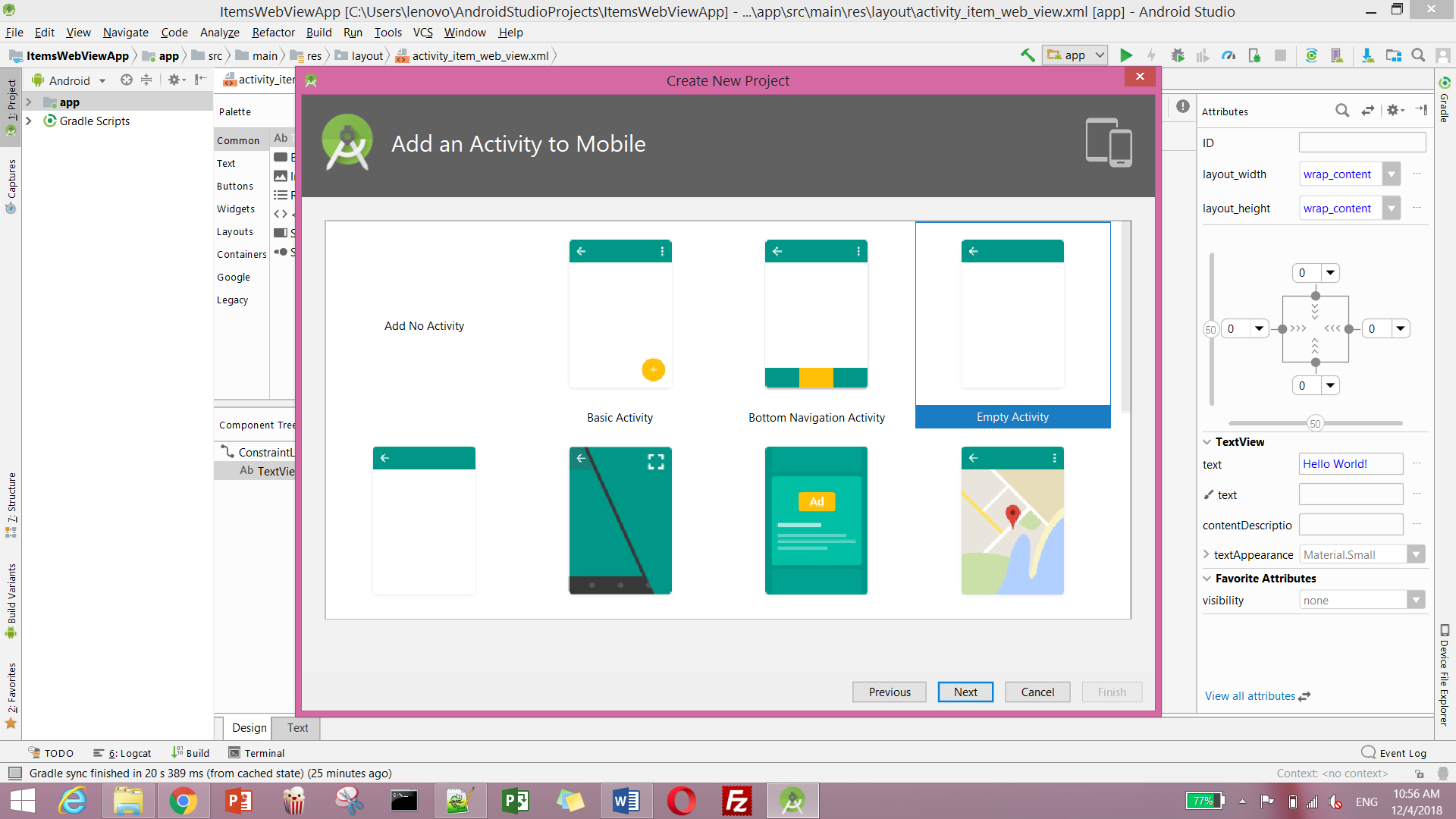




Select an API level ( level 26 Oreo)



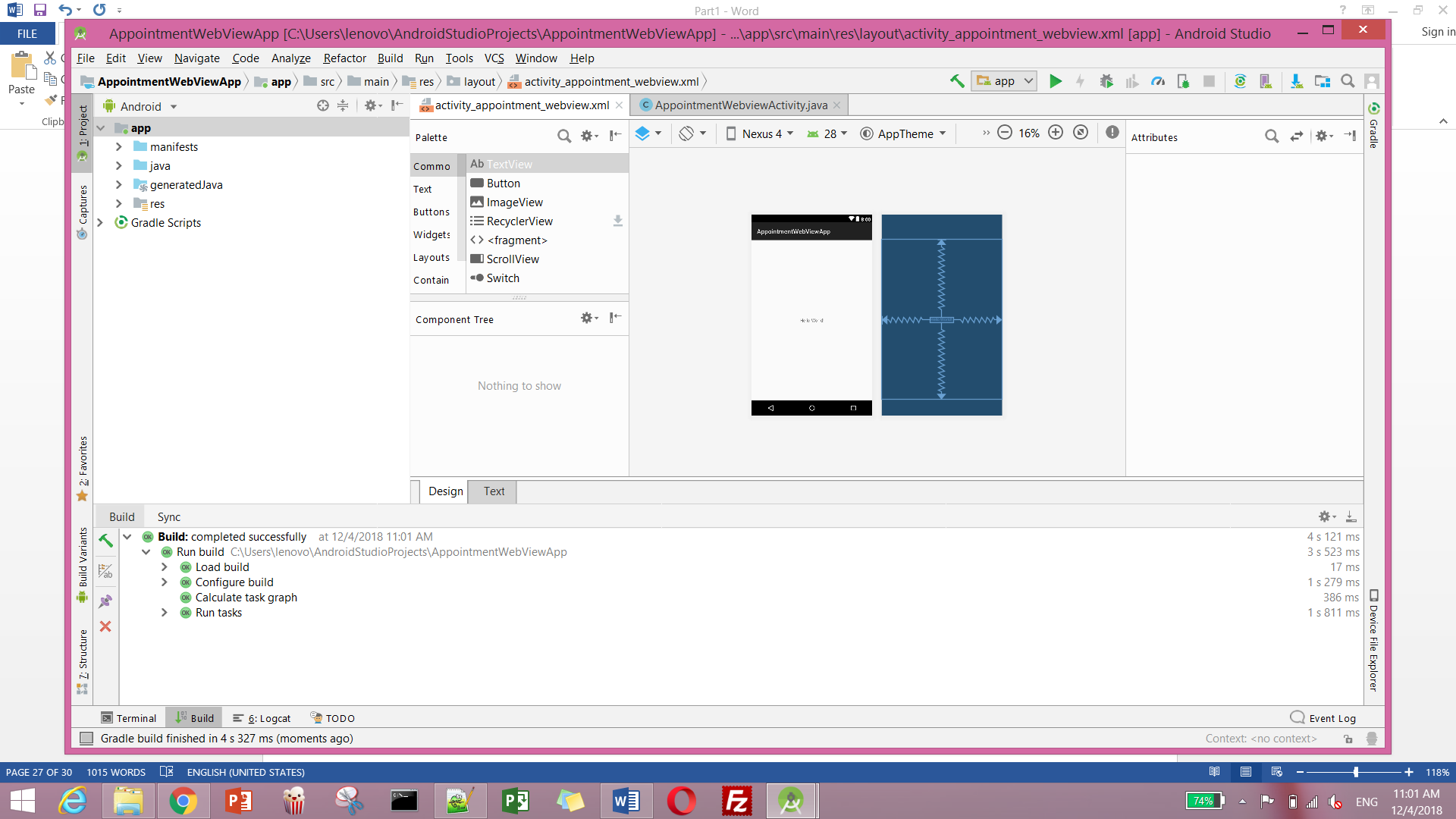
We choose an Empty Activity



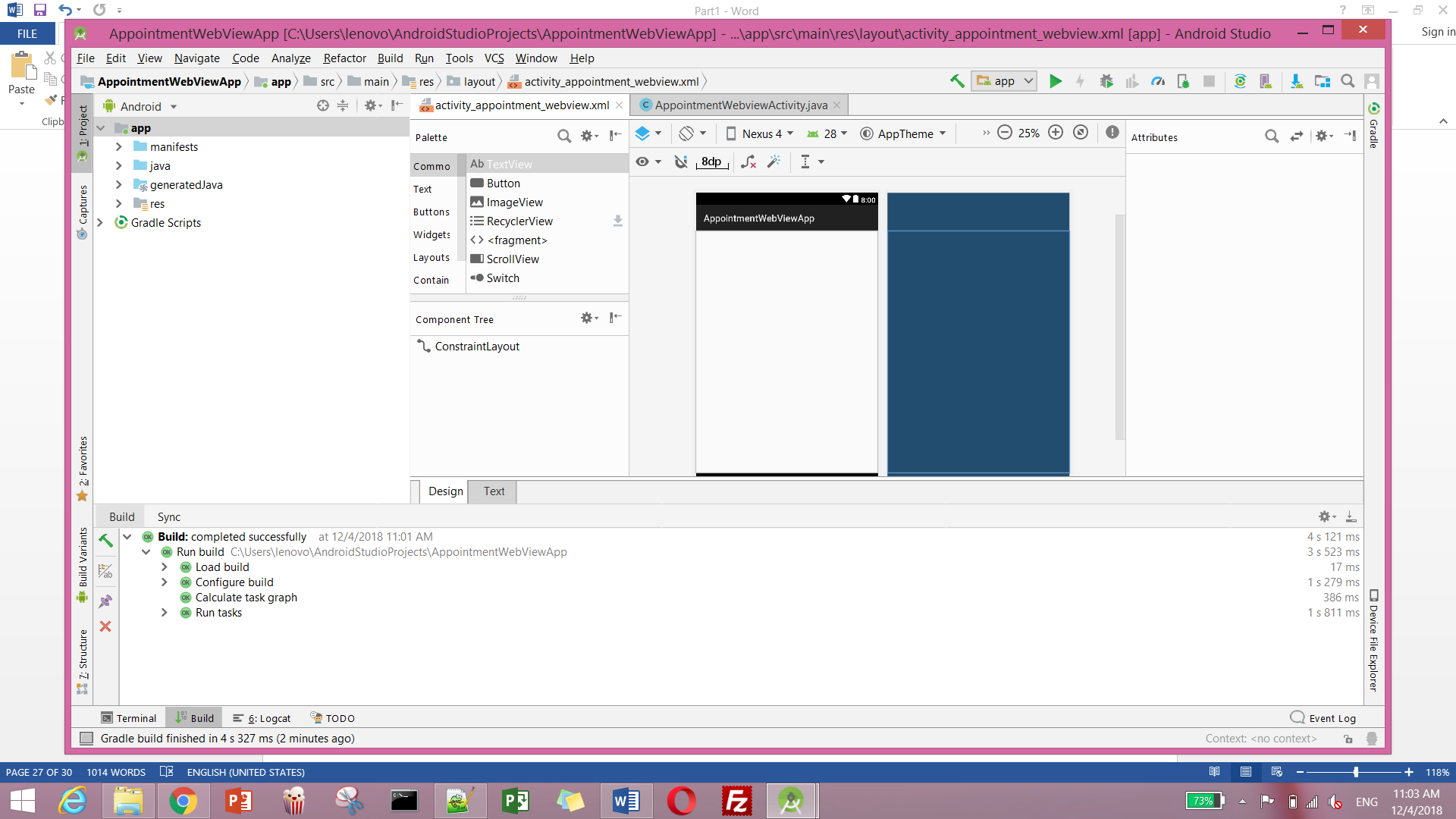
We write name of the activity



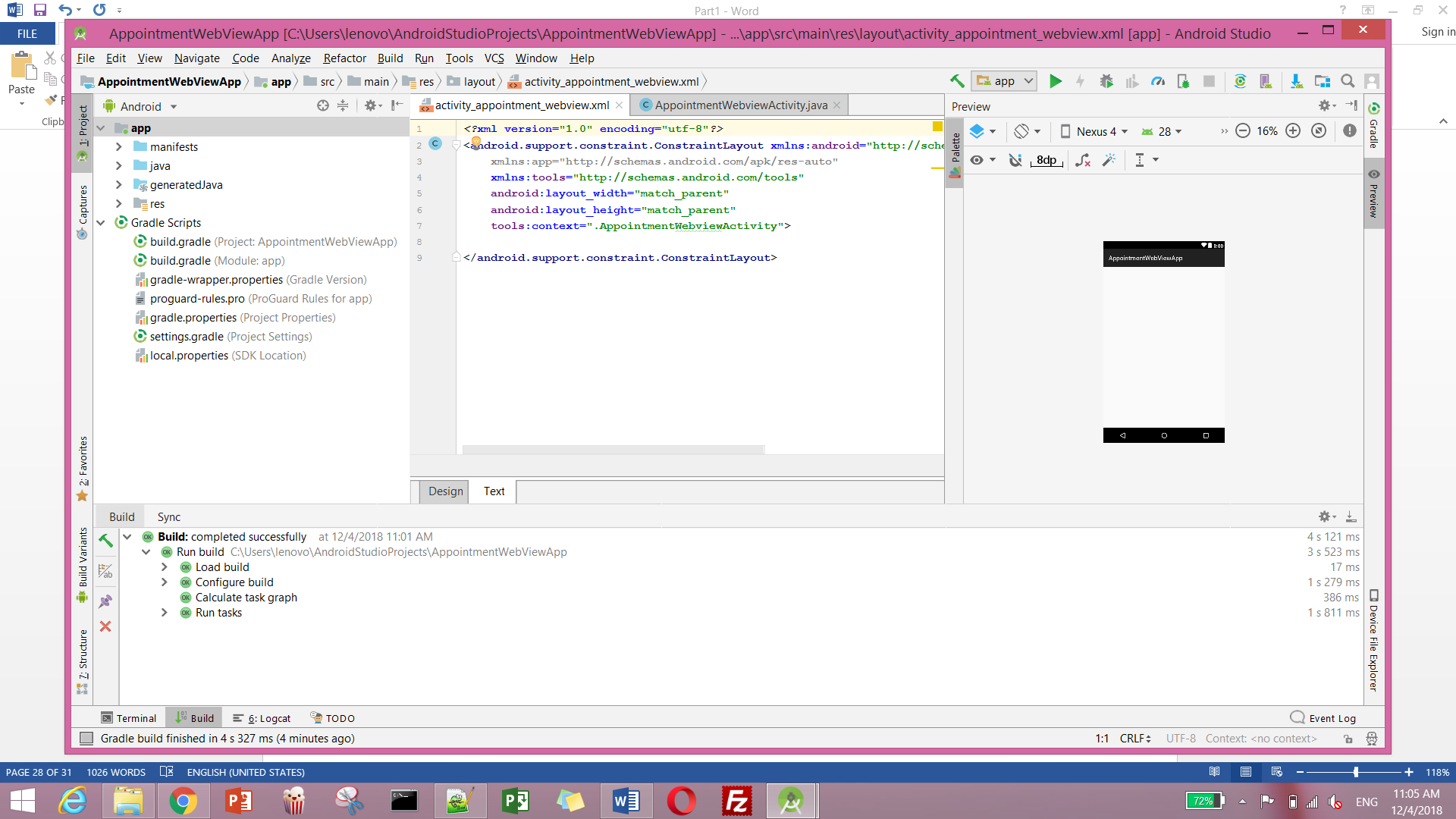
We click on activity XML tab, select the Hello World Text, and delete it



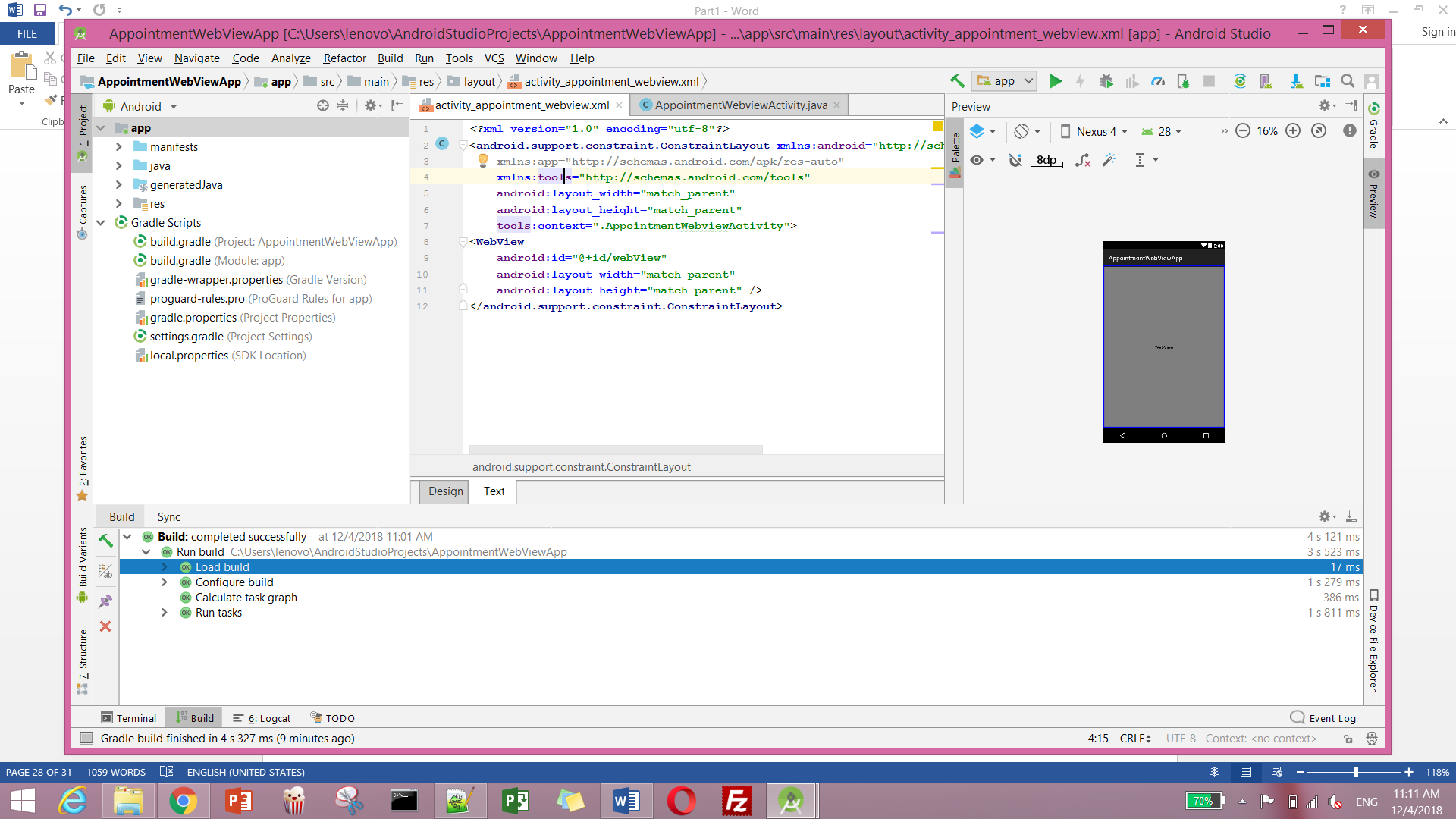
After deleting the design of our app, will looks as shown below



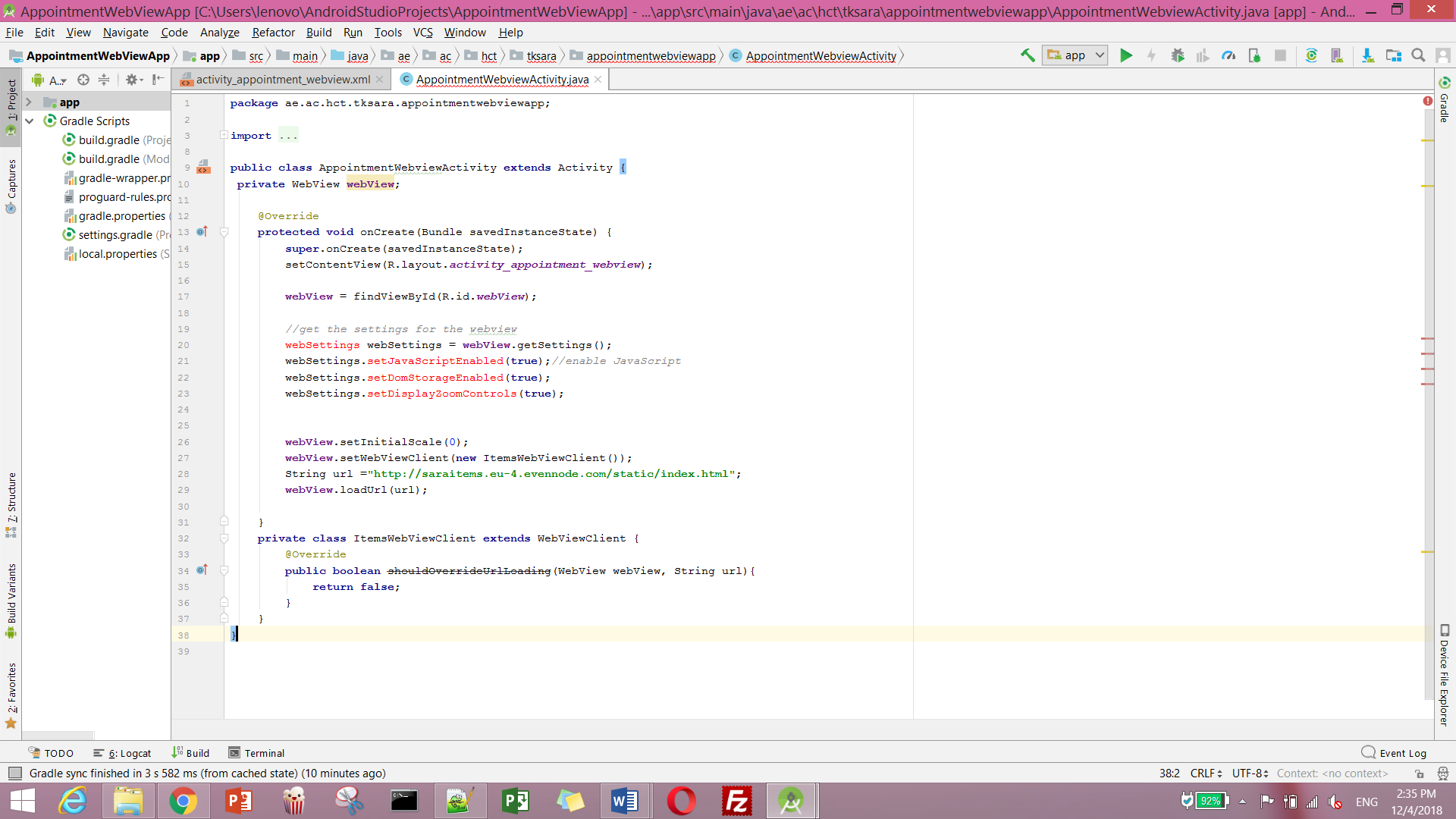
Click on text tab at the button to view the layout XML



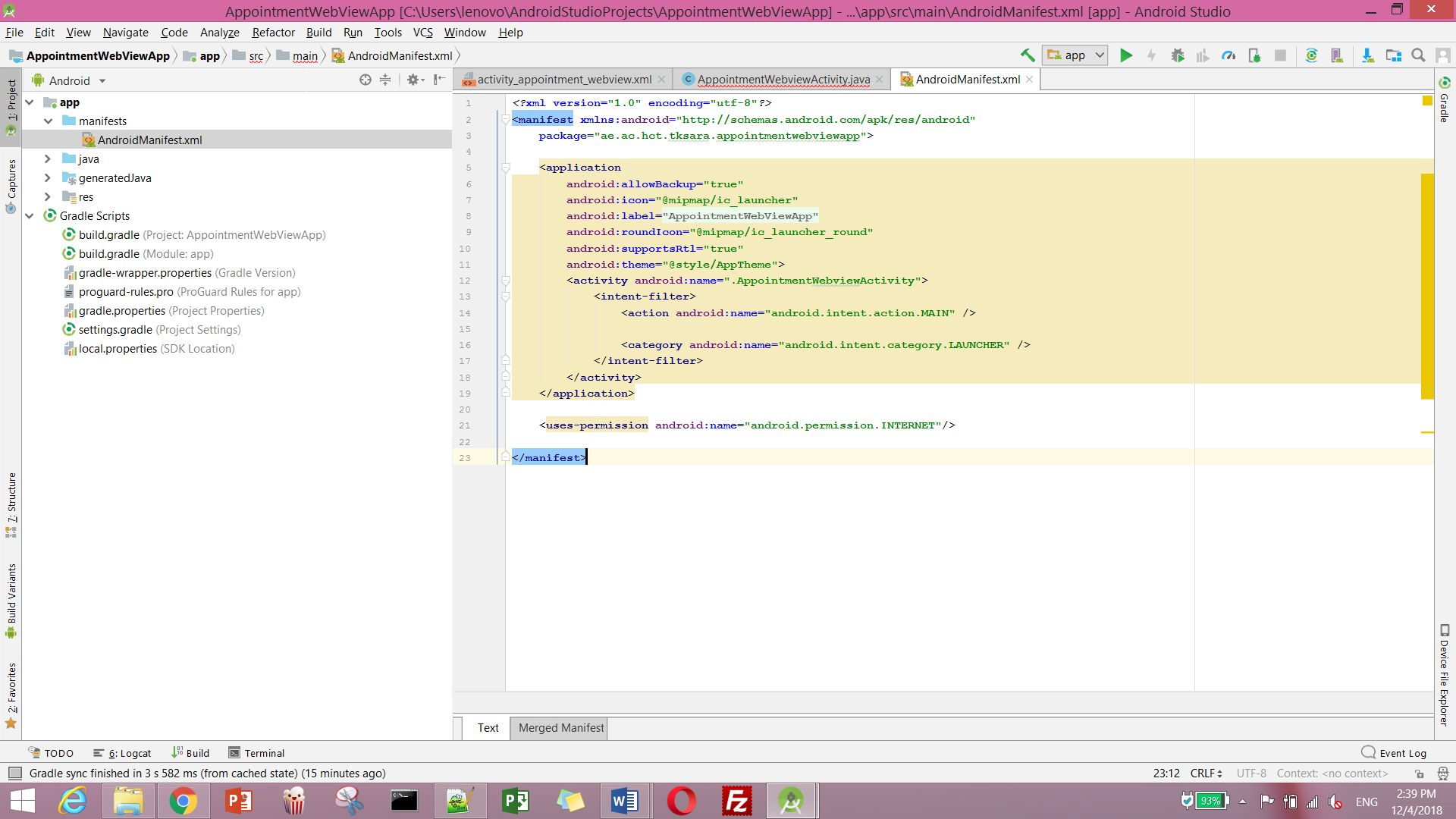
Here we will change the layout of XML from ConstraintLayout to LinearLayout and add a WebView with the attributes as shown



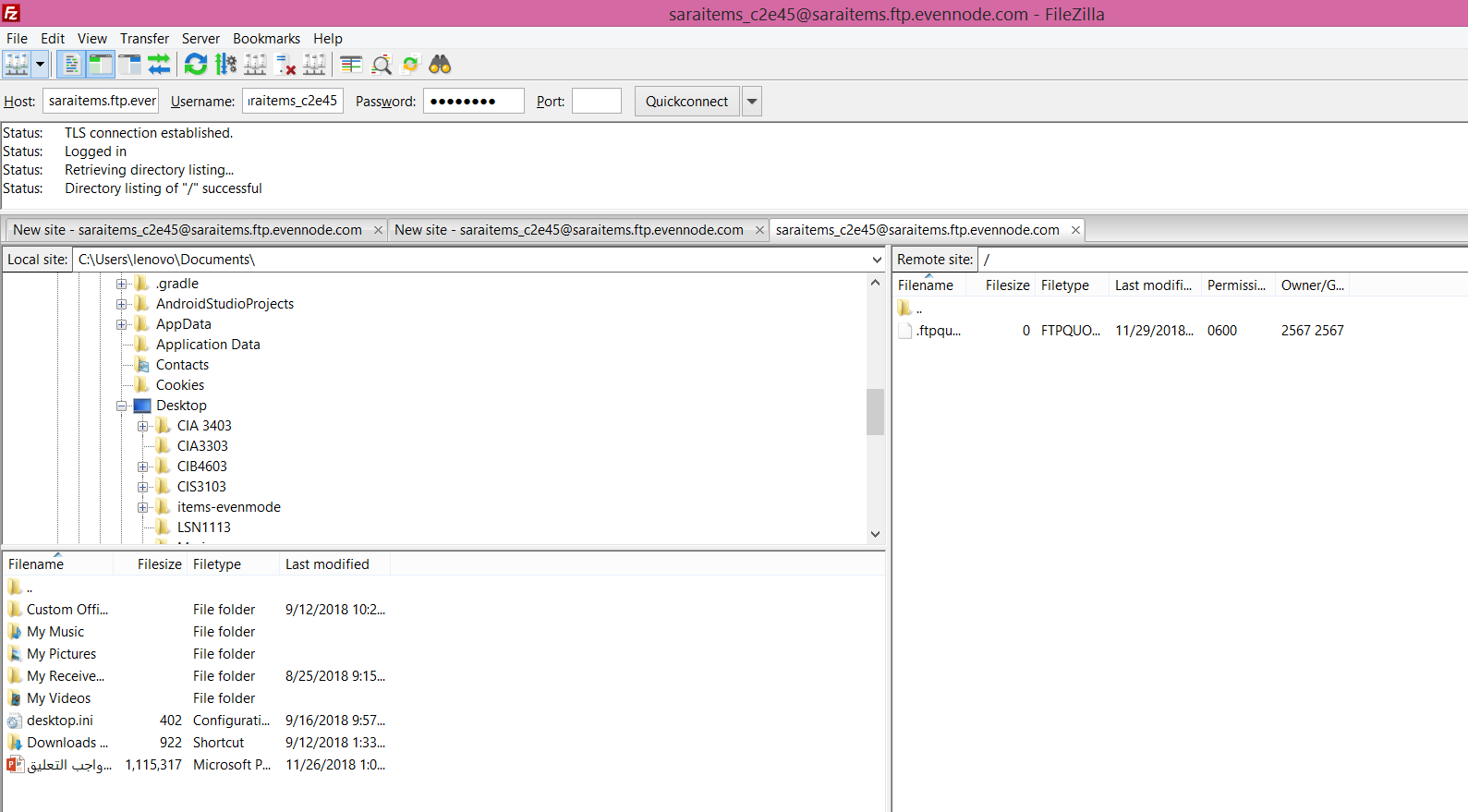
Add the code in the onCreate () as shown below. Also create a private AppointmentWebViewActivity class which is a subclass of AppointmentWebViewActivity and overrides its shouldOverrideUrlLoading () method to false. This will make sure that aby links that clicked on in the app open in our webview and not in an external browser such as Google Chrome.



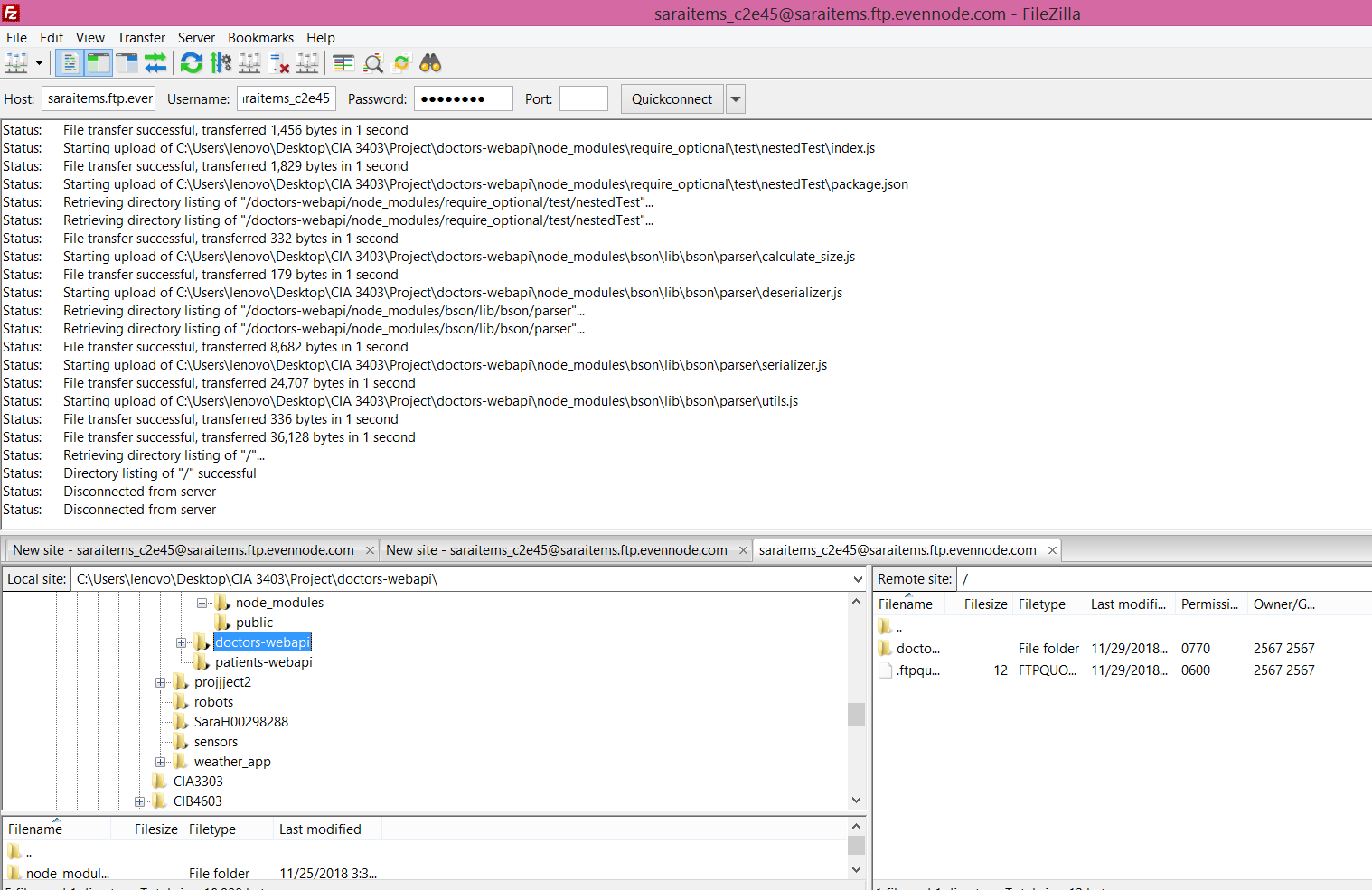
To AndroidManifest.xml file in your apps manifesr folder (Shown on the left) add the user-permission tag for INTERNET permission as shown below.



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*



Uploading doctor-webiap



**Upload App**

