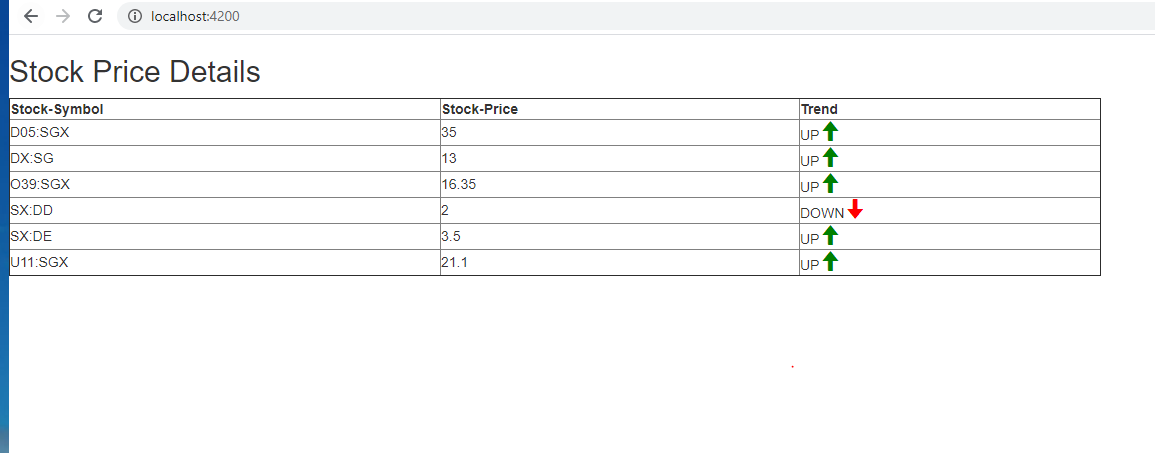
**StockProject-Overview**

**===================**

The stockproject is used to display the stocks in a single web page to the user.It display the stocks dynamically based on the stock changes i.e whenever a new stock is added, it gets reflected instantly without the page reload. Similary whenever there is change in the stock price, the new market price is also reflected showing the trend as, it’s going up or down.

This front page is in Angular [Angular: 9.1.9] and it displays the stocks as below:



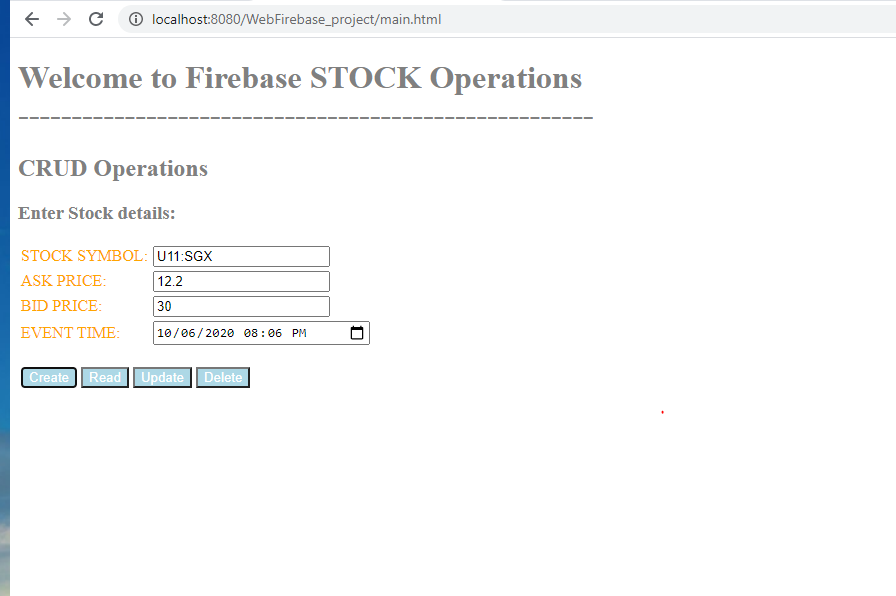
The details are fetched from firebase database which is a nosql realtime DB.when ever there is a change in the table node for the any of the stocks present or if there is a new stock entry or a deletion for the stock, its getting reflected in the frontend immediately without reload of the application. This is one of the major feature of firebase i.e it helps to display the data realtime.From angular, we can subscribe to this database stocklist data and display the same on to the webpage.

Stock data to firebase DB

=====================

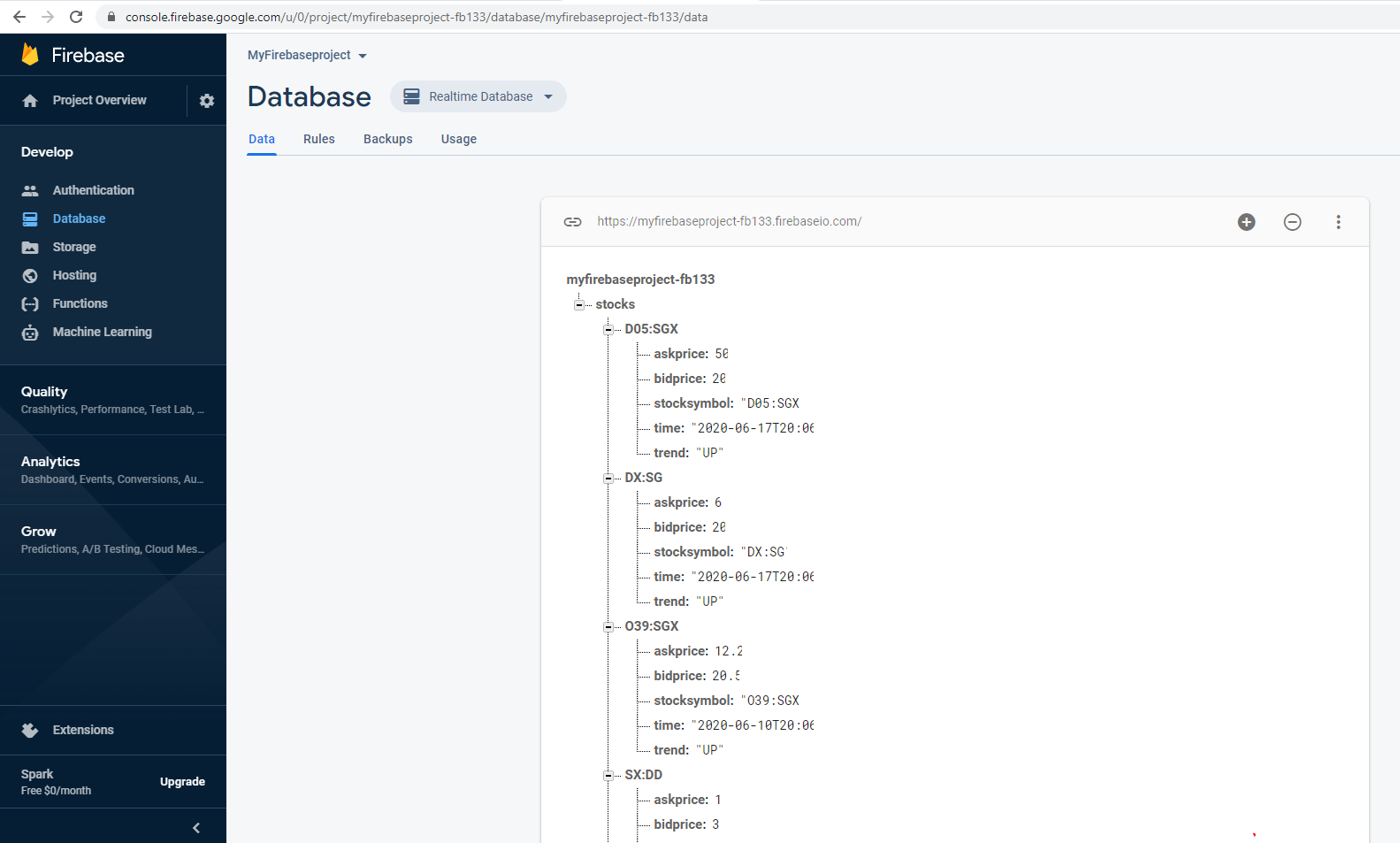
For inserting stock details to the firebase DB, there is a backened webapplication which feeds data for the stock symbols , calculates the market price based on the ask price and bid price and show the trend for the stock as UP or DOWN based on the previous value present in the database stocks table.

The web app for performing DB operations is as below:



This application is running on tomcat at 8080 and helps to load the data into the firebase. The create is used to add a new stock. The update is used to update the stock prices.This checks for existing stocks and updates the trend based on the previous value prevent. The event time is used to reflect the event time for the stock feeds.The delete can be used to remove an existing stock and read helps to fetch the details to the console.

Firebase DB structure created:



Project Setup

**Copy the projects shared under github repository path to local machine:**

<https://github.com/saumeyabobin/StockProjectRepo>

**For frontend**: Copy and unzip the file stockproject.zip and follow the steps under “Front end angular web app setup” provided below in this doc

**For backend**: Use the war [WebFirebaseproject-0.0.1-SNAPSHOT.war](https://github.com/saumeyabobin/StockProjectRepo/blob/master/WebFirebaseproject-0.0.1-SNAPSHOT.war) . The source code for webproject is also shared under the same path. It’s a simple web app and only need to add the firebase config settings into the firebase.js file present in it. Refer to steps under “ Back end webapplication setup” section mentioned below in this doc.

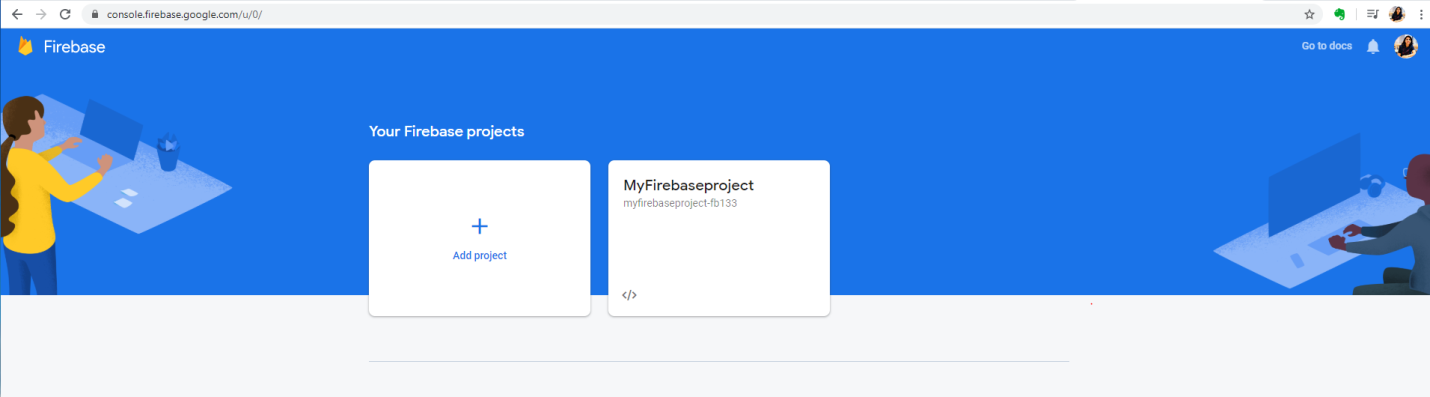
**Database setup**: Refer the firebase database setup below.

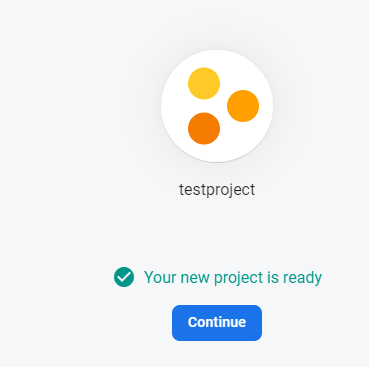
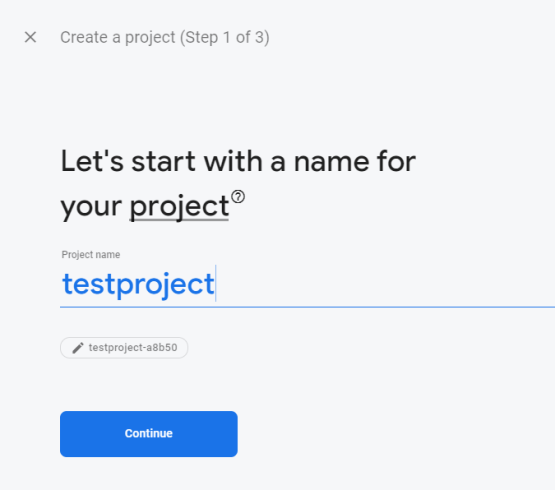
**firebase database setup**

**========================**

**Setup a firebase DB**.[it’s a nosql db which can be setup easily and it’s a realtime database]

Step1: goto firebase.google.com or <https://console.firebase.google.com/> and create a firebaseproject[eg:MyFirebaseproject]

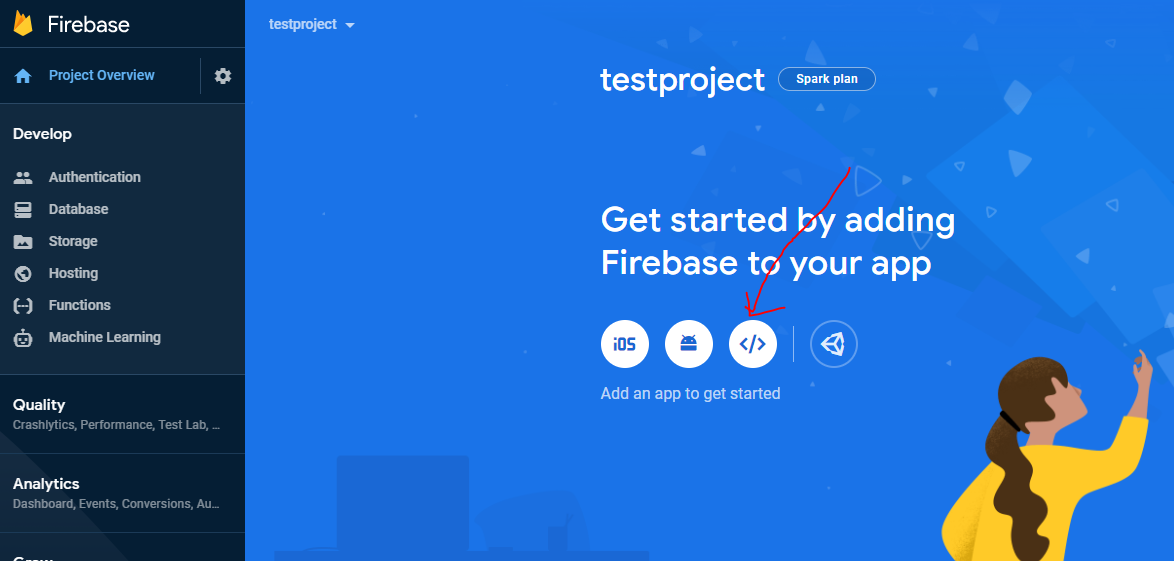


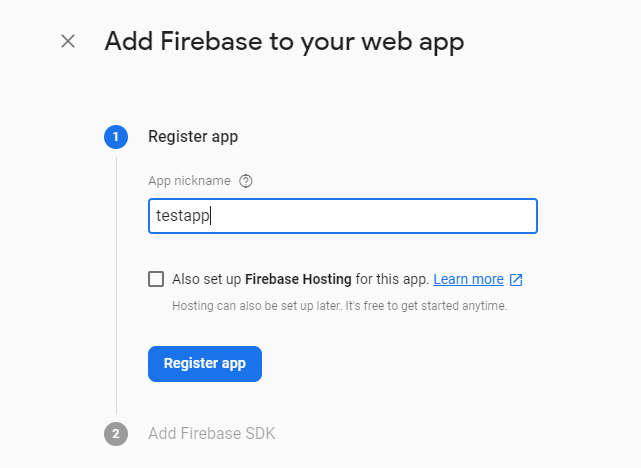


For this project, I have used the project name as MyFirebaseproject. Select the default and press continue and the project will be created.

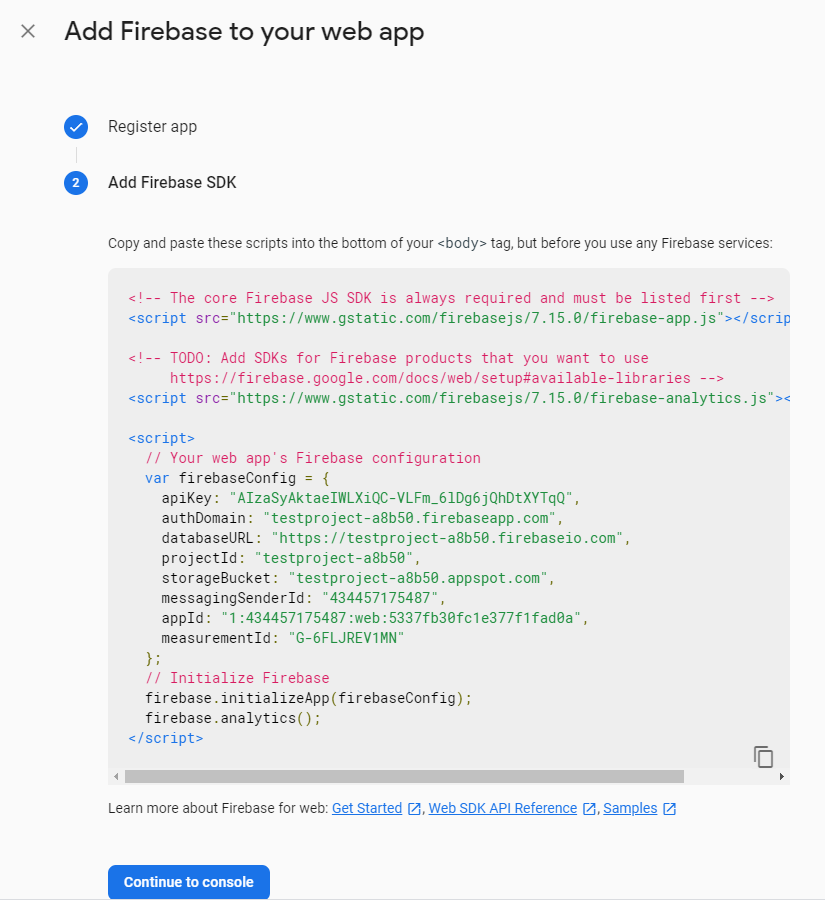
Step2: Setup an app for the project.[eg:stockapp]

Goto project overview page. Based on the type of the app, we need to select the appropriate option. Since we are using a web application to fetch the firebase data, select the web app option to create the app . Add the app name and do the registration to add the app to the project created. **[Please note: The script file generated while creating the app[in step2] is used in the application side to do the initialization for the firebase database connection to access the data.] Copy the firebaseconfig properties to the environment.ts file in stockproject shared.**

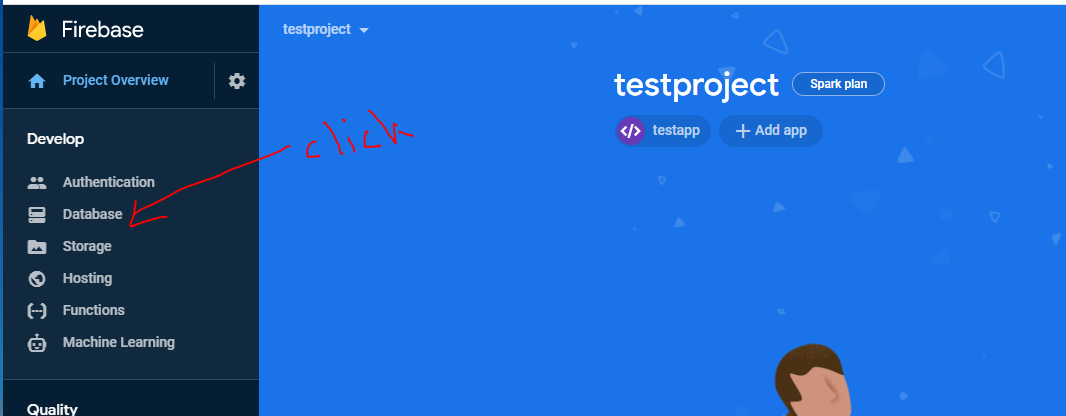




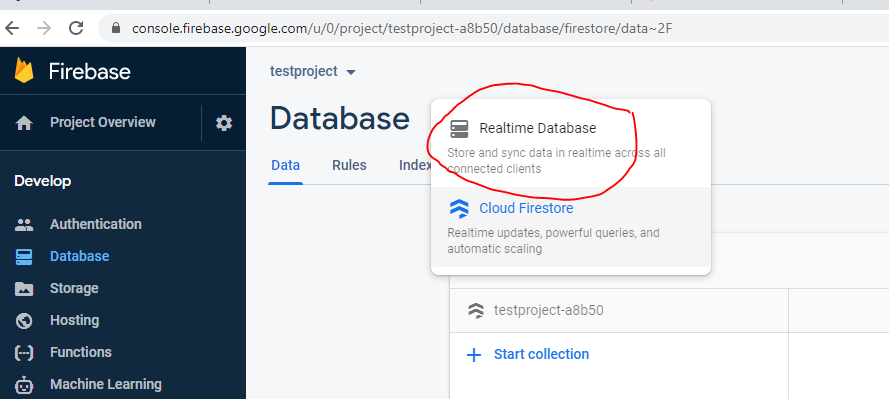
Copy and keep the below config created



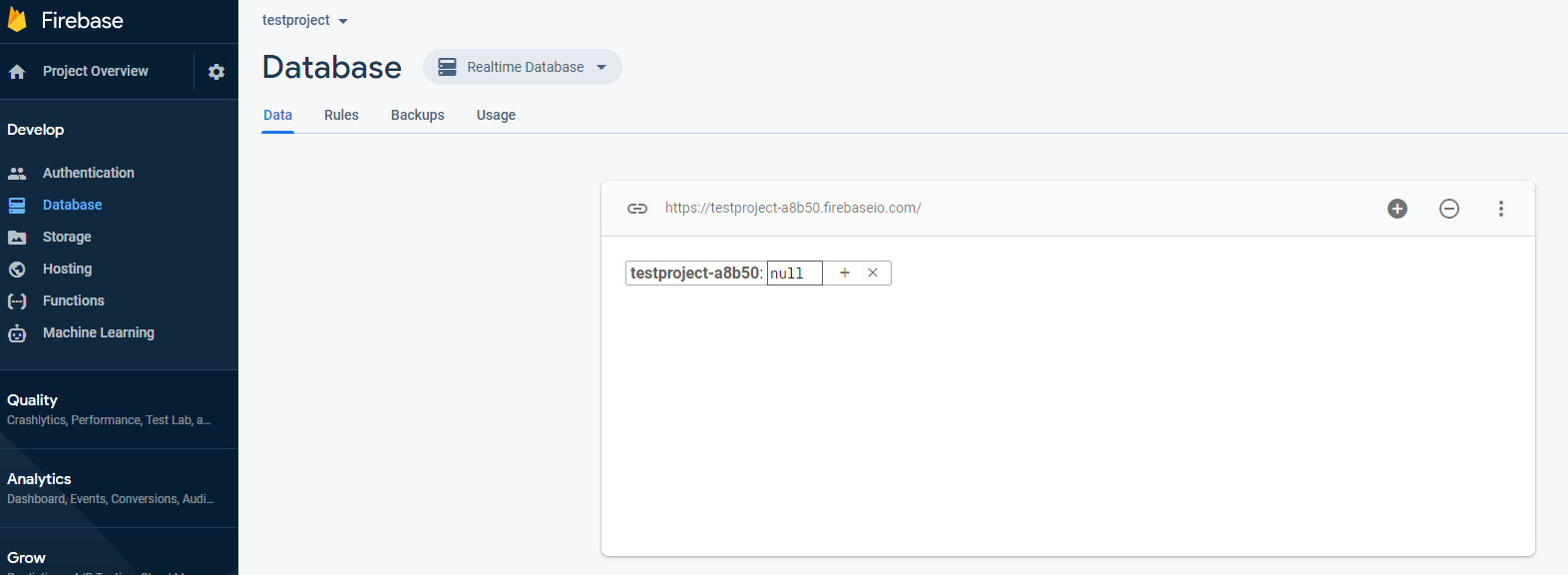
Step3: Create a database for the project.



Select realtime database option



This will create an empty database for the project as below.We can add the nodes and data to it.

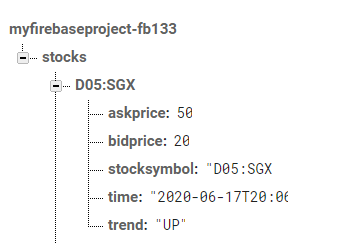


For the current project, the details are as below. It can vary based on the firebase instance created. Based on the config, you can connect to the associated DB, since it provides all the details. **The table name to create is stocks**. Add the child node:D05:SGX and its child elements: askprice,bidprice,stocksymbol,time and trend.[Pls note. This can also be added from the create option from the WebFirebaseproject]

Project name: **MyFirebaseproject**

App name: **stockapp**

Table : **stocks** [Each node in stocks store a stock symbol details. Each node is given a name[eg: DO5:SGX] same as that of the stock symbol. Eg:



Sample Config file generated from the fire base app creation:

<!-- The core Firebase JS SDK is always required and must be listed first -->  
<script src="https://www.gstatic.com/firebasejs/7.15.0/firebase-app.js"></script>  
  
<!-- TODO: Add SDKs for Firebase products that you want to use  
     https://firebase.google.com/docs/web/setup#available-libraries -->  
<script src="https://www.gstatic.com/firebasejs/7.15.0/firebase-analytics.js"></script>  
  
<script>  
  // Your web app's Firebase configuration  
  var firebaseConfig = {  
    apiKey: "AIzaSyA8o3qxzT8RSovx2VmwJN1oOblY0T9ff5s",  
    authDomain: "myfirebaseproject-fb133.firebaseapp.com",  
    databaseURL: "https://myfirebaseproject-fb133.firebaseio.com",  
    projectId: "myfirebaseproject-fb133",  
    storageBucket: "myfirebaseproject-fb133.appspot.com",  
    messagingSenderId: "99563451489",  
    appId: "1:99563451489:web:bc092493e88c271828a620",  
    measurementId: "G-DG2XJLT6LW"  
  };  
  // Initialize Firebase  
  firebase.initializeApp(firebaseConfig);  
  firebase.analytics();  
</script>

**Front end angular web app setup**

**Prerequisites:**

We need nodejs installed and angular cli also installed.

Got [nodejs.org](http://nodejs.org/) and install the latest stable version based on your OS. It provides tool to develop angular project

Goto cmd mode and use the below commands to check for the installation.

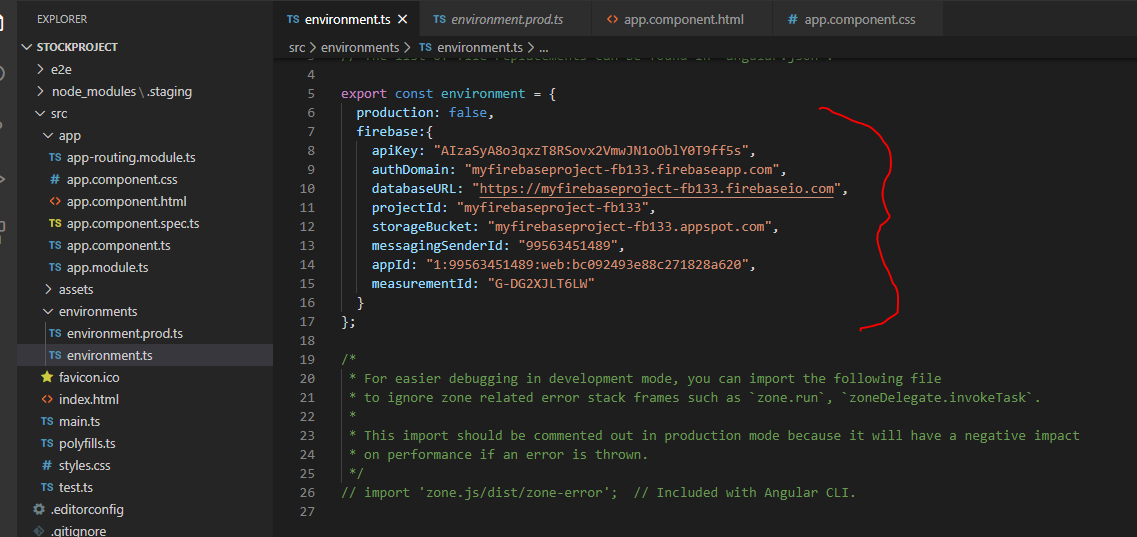
>node --version[get node version installed]

>npm install –g @angular/cli [install angular cli][skip this if already present]

>ng --version [check the angular version]

Copy the shared **stockproject** folder on to destination machine. Then goto this location in command drive.[ If visual studio code is present just use[code .] to open up the project in code editor. Later open terminal window and run the required cmds].

**Important: Copy the firebase properties[refer Step2 in firebase setup: Setup an app for the project] which got created during firebase db setup into environment.ts file under stockproject folder[C:\demo\stockproject\src\environments].**



a)Run the “npm install” command to load the node modules for the project

**Stockproject> npm install**

b)install the firebase modules for the angular

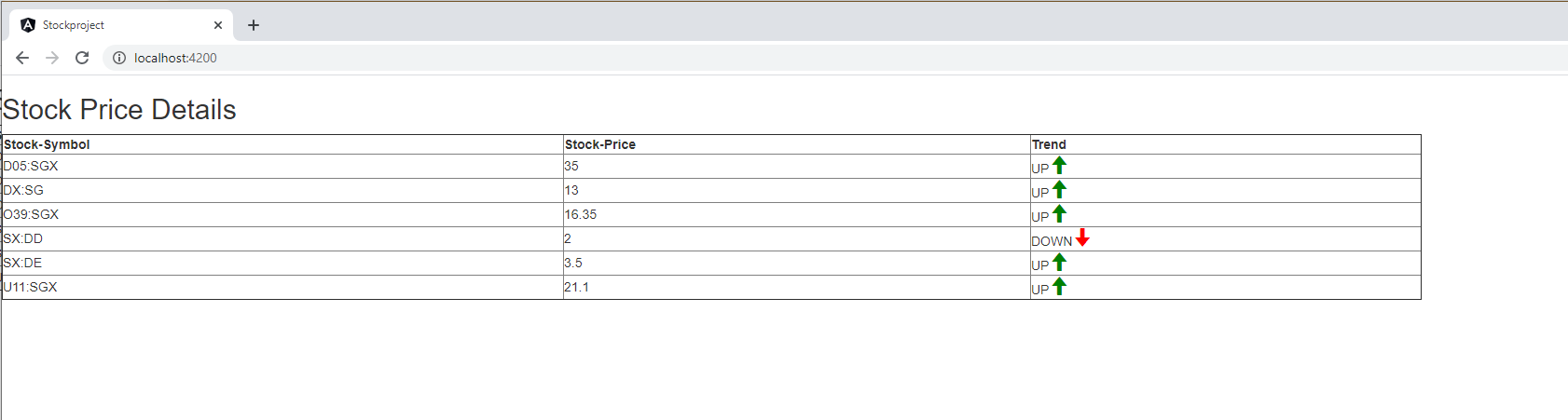
**stockproject>npm install firebase angularfire2 --save**

c)build the project

**Stockproject>ng serve**

The setup is ready . Now load the page from the browser. If there is a firebase db setup present with node called stocks with the required data fields, then it will fetch the data and display as below.

<http://localhost:4200/>

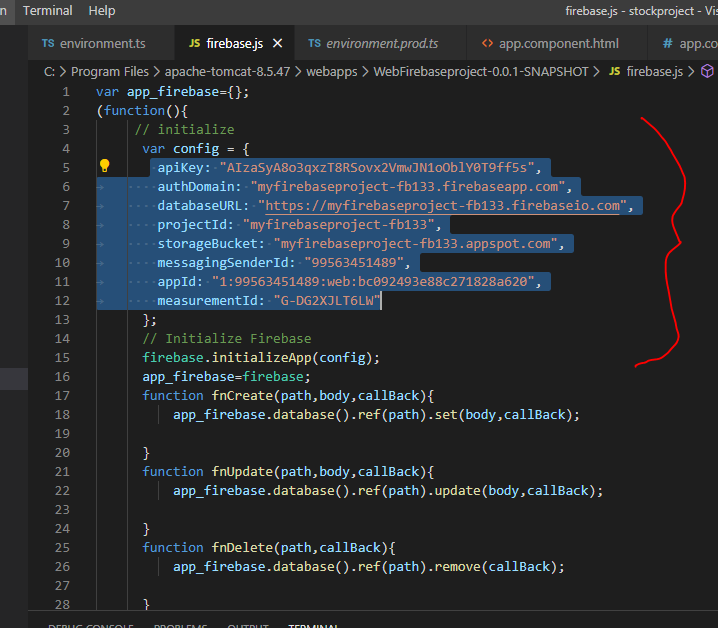


**Back end webapplication setup**

Copy the shared war file[WebFirebaseproject-0.0.1-SNAPSHOT.war] into the apache tomcat webapps folder or any other server of your choice. Start the tomcat. Load the application to see the page loading or not:

<http://localhost:8080/WebFirebaseproject-0.0.1-SNAPSHOT/main.html>

Then edit the firebase config[firebase.js] inside the tomcat webapps folder[C:\Program Files\apache-tomcat-8.5.47\webapps\WebFirebaseproject-0.0.1-SNAPSHOT\firebase.js]. Replace the below config properties with the config file properties generated during firebase app setup[**Pls note: Copy the firebase properties[refer Step2: Setup an app for the project] which got created during firebase projects setup into firbase.js]**



Then restart the tomcat and reload the page again to do the database operations. Goto front end angular app to see the stock changes.

==============================================THE END================================