

Name: *Abdul Manan*

PHP Register:

PHP and mySQL [Register](#) [Login](#)

Registration:

Username

Email

Password

Password

Confirm Password

Confirm Password

Address 2

Apartment, studio, or floor

City

Zip

Sign in

Figure 1 PHP Registration Page

PHP Login:

Php Login System [Register](#) [Login](#)

Please Login Here:

Username

Enter Username

Password

Enter Password

Submit

Figure 2 PHP Login Page

MySQL Database:

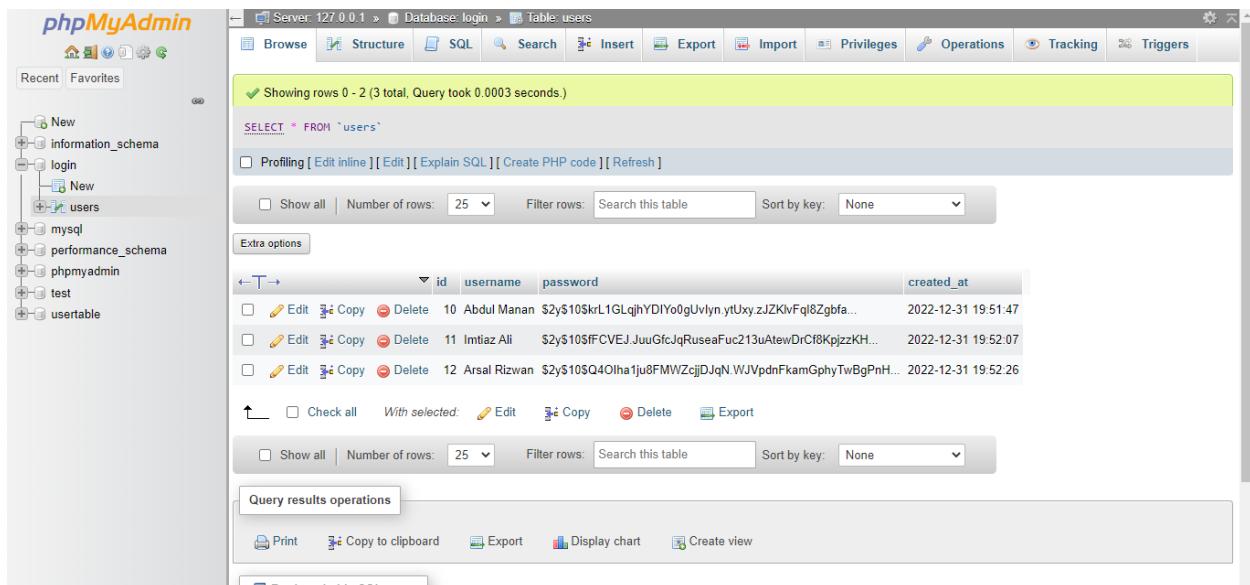


Figure 3 SQL Database

WEBPAGE:

After entering the correct credentials, we can access the webpage. We made a webpage over HTML, CSS, PHP and JavaScript where Professor can enter final result of a student over the website.

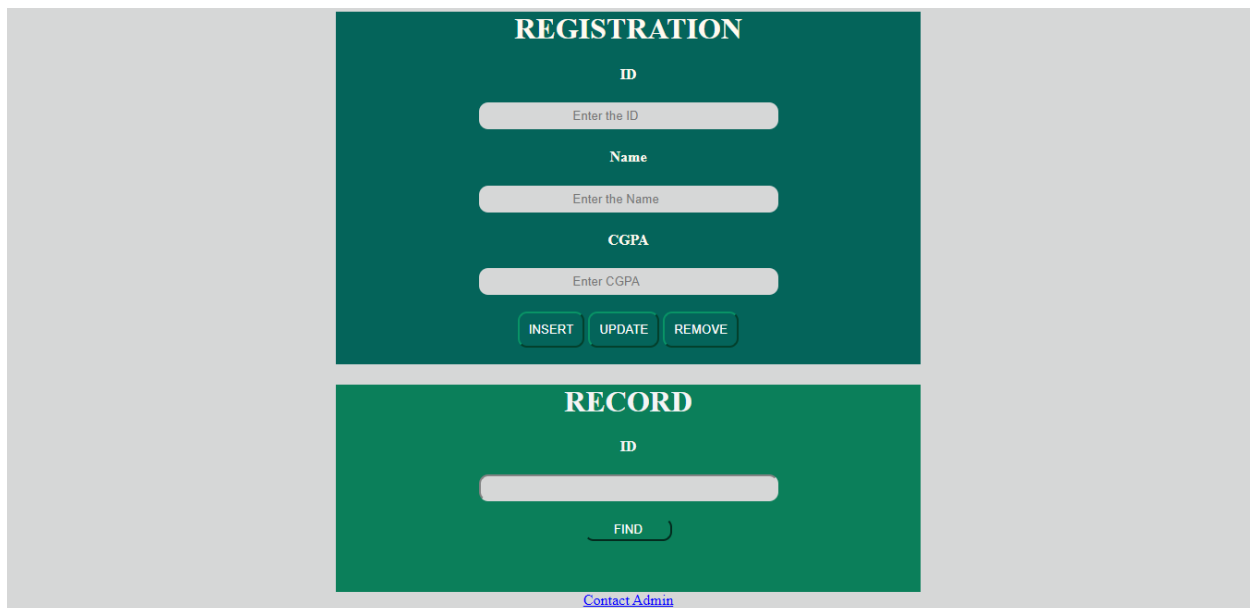


Figure 4 Front End of Website



The image shows a registration form with a dark green background and white text. At the top, the word "REGISTRATION" is centered in a large, bold, serif font. Below it, the label "ID" is centered. A light gray rounded rectangular input field contains the number "5". Below this, the label "Name" is centered. Another light gray rounded rectangular input field contains the name "Danial". Below that, the label "CGPA" is centered. A third light gray rounded rectangular input field contains the value "3.65". At the bottom, there are three rounded rectangular buttons with green borders and white text: "INSERT", "UPDATE", and "REMOVE".

REGISTRATION

ID

5

Name

Danial

CGPA

3.65

INSERT UPDATE REMOVE

Figure 5 Entering Data for a Student



The image shows a record view with a dark green background and white text. At the top, the word "RECORD" is centered in a large, bold, serif font. Below it, the label "ID" is centered. A light gray rounded rectangular input field contains the number "5". Below this, a rounded rectangular button with a green border and white text contains the word "FIND". Below the button, the text "Name: Danial" is centered. At the bottom, the text "CGPA: 3.65" is centered.

RECORD

ID

5

FIND

Name: Danial

CGPA: 3.65

Figure 6 Checking the successful storage of Data

```

<script type="module">

  // Import the functions you need from the SDKs you need
  import { initializeApp } from "https://www.gstatic.com/firebasejs/9.14.0/firebase-app.js";
  // TODO: Add SDKs for Firebase products that you want to use
  // https://firebase.google.com/docs/web/setup#available-libraries

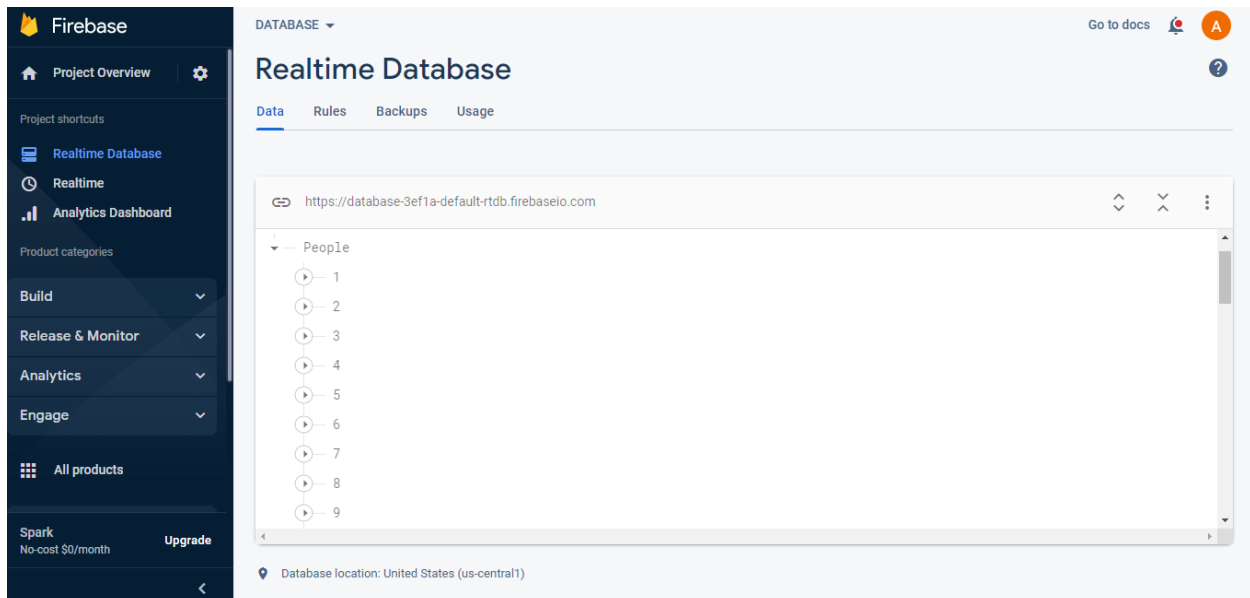
  // Your web app's Firebase configuration
  const firebaseConfig = {
    apiKey: "AIzaSyA0DdnMW_tDgJLd5BZX2d1WdWkqs-BbcY8",
    authDomain: "pro3-1df7b.firebaseio.com",
    databaseURL: "https://database-3ef1a-default-rtdb.firebaseio.com/",
    projectId: "pro3-1df7b",
    storageBucket: "pro3-1df7b.appspot.com",
    messagingSenderId: "884996298701",
    appId: "1:884996298701:web:0aa6561650bbd3c89dd684"
  };

  // Initialize Firebase
  const app = initializeApp(firebaseConfig);

```

Figure 7 Connecting Firebase APIs

FIREBASE:



The screenshot shows the Firebase Realtime Database interface. On the left is a sidebar with navigation options: Project Overview, Realtime Database, Realtime, Analytics Dashboard, Build, Release & Monitor, Analytics, Engage, All products, and Spark. The main area displays the 'Realtime Database' for the project 'DATABASE'. The 'Data' tab is selected, showing a tree view of the database structure. The root node is 'People', which contains a list of 9 items, each represented by a circle with a number inside. The URL bar shows the database URL: 'https://database-3ef1a-default-rtdb.firebaseio.com'. At the bottom, it indicates the database location: 'United States (us-central1)'.

Figure 8 Storing data for long term record on Firebase Realtime Database

Grades

ID ▲	Name ▲	CGPA ▲
1	Ali	3.66
2	manan	3.33
3	Talha	3.5
4	Imtiaz	3.5
5	Danial	3.55
6	Haider	3.44
7	Sadiq	3.77
8	Kanwar	3.65
9	Anas	3.9
10	Zain	3.22

Figure 11 Students can see the data through this Front End Visible to them

Professors who entered your CGPA

ID ▲	Professors ▲
1	Admin
2	Waqas Ahmad
3	Arsal Rizwan
4	Imtiaz Ali
5	Abdul Manan

Figure 12 Students can see all the professors who entered their data

12/25/2022, 6:14:14 PM node: debug 13

msg.payload : Object

```
▼ object
  ▼ People: array[51]
    ▼ [0 ... 9]
      0: undefined
      1: object
      2: object
      3: object
      4: object
      ▼ 5: object
        Age: "3.65"
        ID: "5"
        Name: "Danial"
      6: object
      7: object
      8: object
      9: object
    ▶ [10 ... 19]
    ▶ [20 ... 29]
    ▶ [30 ... 39]
    ▶ [40 ... 49]
    ▶ [50 ... 50]
```

Figure 13 Data is stored in msg payload through this format

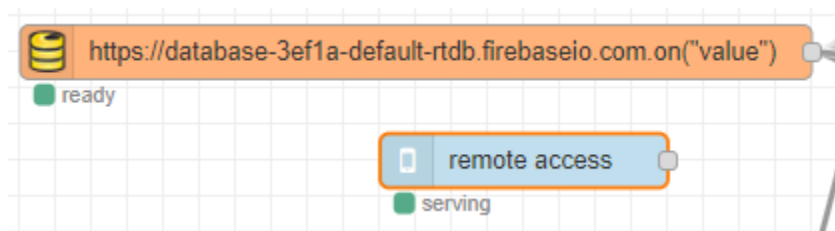


Figure 14 Nodes Used for Firebase and Remote Connection

Edit remote-access node > **Edit remote-config node**

Delete Cancel Update

Properties

Name Node-REDD

This name will be displayed in the app.

Serving Host localhost

localhost = Host of Node-RED, 10.25.40.20 for internal use

Protocol http

Serving Port 1880

Base URL /ui

Server location Europe (Germany)

Figure 15 Configuration of Remote Access



Figure 16 Multiple devices could be connected to access data through barcode

Contact Form:

Contact Us!

Name

Email

Phone

How can we help you?

Figure 17 Contact Form by Professors to contact admin that send data to Gmail and Excel

Primary

Promotions 48 new
Instructables, Alison Courses, M...

Social 25 new
Nadir Mushtaq via LinkedIn, Link...

☐ ☆ me

Contact form submitted - name Danial email Danial@gmail.com phone 03213274831 message I need t...

6:06 PM

Figure 18 Data in Gmail

Contact form submitted Inbox x



abdulmananchaudary@gmail.com

to me ▾

Name

Danial

Email

Danial@gmail.com

Phone

03213274831

Message

I need to make a report of it

← Reply

→ Forward

Figure 19 Gmail Data

The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D	E	F
1	Timestamp	name	email	phone	message	
2	12/25/2022	Danial	Danial@gmail.com	3213274831	I need to make a report of it	

Figure 20 Data on Excel Sheet

Conclusion:

We made SRA result portal for university that manages multiple databases using IOT platform NODE-RED. Main motive is to make ease for people and administration to easily access and manage data thoroughly. We learned about the connection of multiple interfaces through extensions and APIs and their uses. We learned multiple solutions for communication in between devices to manage data thoroughly.